Mitigating Adverse Effects of Using Online Social Networks

Verminderung negativer Effekte bei der Nutzung von Online Social Networks
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von Dipl. Wirt.-Inf. Thomas Paul aus Mühlhausen
Tag der Einreichung: 2. November 2015, Tag der Prüfung: 16. Dezember 2015,
Erscheinungsjahr: 2016
Darmstadt — D 17

1. Gutachten: Prof. Dr. Ing. Thorsten Strufe
2. Gutachten: Prof. Dr. Ing. Wolfgang Effelsberg
3. Gutachten: Prof. Dr. rer. nat. Chris Biemann

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
P2P Networks

Mitigating Adverse Effects of Using Online Social Networks
Verminderung negativer Effekte bei der Nutzung von Online Social Networks

Genehmigte Dissertation von Dipl. Wirt.-Inf. Thomas Paul aus Mühlhausen

1. Gutachten: Prof. Dr. Ing. Thorsten Strufe
2. Gutachten: Prof. Dr. Ing. Wolfgang Effelsberg
3. Gutachten: Prof. Dr. rer. nat. Chris Biemann

Tag der Einreichung: 2. November 2015
Tag der Prüfung: 16. Dezember 2015,
Erscheinungsjahr: 2016

Darmstadt — D 17

Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den May 28, 2016

_____

(Thomas Paul)

Abstract

Online Social Networks (OSNs), such as Facebook and Google+, are network-based communication systems. They allow their users to create persistent digital representations of themselves, called user profiles, and to establish explicit connections to other user profiles within a bounded system. OSNs have become popular services that attract more than one billion users. The ubiquity of mobile devices, which support accessing OSNs from anywhere, causes OSNs to be powerful communication tools that change the way how people interact with each other. OSNs gain importance for the daily life of their users by becoming an access channel to parts of their social environment.

However, using OSNs puts users at the risk of various undesired side-effects, such as cyber bullying, being forced to grant content use licenses to OSN providers, facilitating social engineering attacks, losing the job due to improper public comments and even imprisoning in autocratic countries. In addition, users of OSNs are subject to crowd engineering[1]. These side-effects are caused by both: sharing data with too many or the wrong recipients and implications of the fact that OSNs are operated by commercial companies.

This thesis contributes to understand the origin of these side-effects and approaches to mitigate the latter without limiting the power of the OSN. We elaborate user behavior in the most popular OSN, Facebook, and introduce a color-based interface that simplifies audience selection in OSNs. Furthermore, we improve decentralized OSNs (DOSNs) that are not operated by commercial OSN providers to avoid ownership implications.

Our first step is to gain understanding of the sources of adverse effects by evaluating OSN usage in detail. We answer the questions: How intensive do users use the most popular OSN, Facebook? How do users orchestrate the variety of communication functions? With whom do users communicate? On the way to understand causes of side-effects, a second step is to investigate the content sharing preferences amongst users from various countries. Since users are unable to choose their audience properly, evaluating the actual settings is insufficient. To circumvent this issue, we developed a browser plug-in that simplifies the audience selection.

Decentralizing OSNs is a way to avoid implications of commercial companies being the owner and operator of the OSNs. In spite of their potential to avoid adverse ef-

[1] http://firstmonday.org/ojs/index.php/fm/article/view/4901/4097, accessed 2015-07-13

fects, DOSNs are not widely adopted. They suffer from the lack of performance and functionality, compared with their popular counterparts. We contribute to mitigate the lack by surveying the state-of-the-art in the field of DOSNs, introducing a search mechanism that allows to find user handles without leaking sensible information, allowing P2P-based DOSNs to be operated under high churn and by prefetching videos based on locally-available information.

Zusammenfassung

Online Social Networks (OSNs), wie z.B. Facebook oder Google+, sind netzwerkbasierte Kommunikationssysteme, die es innerhalb eines geschlossenen Systems ermöglichen Nutzerprofile zu erstellen, und diese mittels expliziten Freundschaftserklärungen zu verbinden. OSNs sind derzeit sehr beliebt. Sie haben mehr als eine Milliarde Nutzer. Die Ubiquität von mobilen netzwerkfähigen Endgeräten macht OSNs zu mächtigen Kommunikationswerkzeugen, die die Art der Kommunikation im Alltag ihrer Nutzer ändern, da sie es erlauben, OSNs unabhängig vom Aufenthaltsort des Nutzers zu verwenden.

Die Nutzung von OSNs birgt jedoch das Risiko unerwünschter Nebenwirkungen. Das Spektrum der möglichen Nebenwirkungen umfasst z.B. Mobbing, das Gewähren von Nutzungsrechten an Kommunikationsinhalten, das Fördern von Social Engineering Angriffen, der Verlust des Arbeitsplatzes und sogar Verhaftung in autokratisch regierten Ländern. Nutzer von OSNs sind außerdem Versuchen ausgesetzt die öffentliche Meinung zu beeinflussen[2]. Diese Nebenwirkungen werden dadurch verursacht, dass Nutzer Informationen mit den falschen bzw. mit zu vielen Rezipienten teilen, sowie durch den Umstand dass OSNs von einem einzigen Akteur - einem Unternehmen - betrieben werden.

Das Ziel dieser Arbeit ist die Verminderung und Vermeidung von Nebenwirkungen die sich aus der Nutzung von OSNs ergeben. Zu diesem Zweck darf die Funktionsvielfalt der OSNs jedoch nicht eingeschränkt werden, um deren Attraktivität für die Nutzer nicht zu gefährden. Dieses Ziel wird sowohl durch ein neuartiges Inferface zur Vereinfachung der Privatsphäreneinstellungen, als auch durch die Verbesserung alternativer, dezentraler OSNs erreicht.

Der erste Schritt dieser Arbeit ist das Erforschen der genauen Ursachen der unerwünschten Nebenwirkungen, indem das Nutzerverhalten untersucht wird. Dabei werden die folgenden Fragestellungen betrachtet: Wie intensiv nutzen Nutzer das derzeit populärste OSN, Facebook? Wie orchestrieren sie die angebotene Funktionsvielfalt? Mit wem kommunizieren die Nutzer? Des Weiteren wurden die Vorlieben der Facebooknutzer bzgl. des Teilens von Inhalten untersucht. Da viele Nutzer nicht in der Lage sind das originale Interface zum Einstellen der Sichtbarkeit von Inhalten fehlerarm zu nutzen, reicht es nicht aus die aktuellen Privatsphäreneinstellungen von Face-

[2] http://firstmonday.org/ojs/index.php/fm/article/view/4901/4097, Zugriff am 13.07.2015

booknutzern zu untersuchen. Statt dessen wurde ein neues, auf einer ampelähnlichen Farbkodierung basierendes Interface in Form einer Browsererweiterung veröffentlicht und anschließend zu Studienzwecken verwendet.

Ein Weg um Auswirkungen des Umstandes, dass OSNs von kommerziellen Unternehmen betrieben werden, zu vermeiden ist das Dezentralisieren von OSNs. OSNs, die nicht von einer einzigen Organisation betrieben werden und auf einer dezentralen technischen Infrastruktur basieren, werden in dieser Arbeit DOSNs genannt. Diese sind derzeit jedoch kaum verbreitet, weil sie gegenüber den heutzutage populären OSNs im Funktionsumfang und der Performanz zurück stehen.

In dieser Arbeit werden daher ausserdem Ansätze vorgestellt, die dabei helfen die Nachteile von DOSNs gegenüber den OSNs zu verringern. Dazu wird zuerst der Stand der Forschung untersucht und anschließend werden drei Beiträge zur Verbesserung von DOSN vorgestellt. Der erste Beitrag ist ein Algorithmus der das Finden von Adressen anderer Nutzer erlaubt, ohne dabei Daten der gesuchten Person Preis zu geben. Der zweite Beitrag verbessert die Verfügbarkeit von Profildaten in P2P-basierten DOSNs unter herausfordernden Bedingungen. Der letzte Beitrag dieser Arbeit untersucht das Laden von Inhalten in DOSN aufgrund von Prognosen, die ausschließlich mithilfe von lokal beim Nutzer verfügbaren Daten erstellt wurden.

Acknowledgements

This thesis would not exist without the support from my advisors as well as my colleagues, students, family, and friends.

In particular, I thank my professor and advisor Prof. Dr. Thorsten Strufe for both: offering the opportunity to me and enabling me to do this dissertation. I thank him for his faith, his support as well as for the opportunities to work with international partners at NTU in Singapur, KTH in Stockholm and Telecom Paritech in Paris. Furthermore, I thank Prof. Dr. Wolfgang Effelsberg for being my co-referee and for supervising me in the MAKI project. In addition, I thank Prof. Dr. Chris Biemann for being the third reviewer of this thesis and Prof. Rolf Hoffmann for both: offering a seat at his group's office to me and for integrating me in his group in my first days at TU-Darmstadt.

I am deeply grateful to my colleagues Dr. Hani Salah, Dr. Paul Gebelein, Andreas Höfer and Stefanie Roos to help me out of mental blocks and I thank all my colleagues in the P2P group and the MAKI project for inspiring discussions. I also thank my office mates Dr. Patrick Ediger and Dr. Christian Schäck for their valuable advices.

This thesis would not be the same without the students who worked under my supervision. I would like to express my deep gratitude to Daniel Puscher who worked more than three years (Bachelor thesis, student helper and Master thesis) on evaluating the C4PS interface and the user behavior in Facebook. Niklas Lochschmidt implemented and evaluated Lilliput in his Master thesis and Marius Hornung implemented and evaluated the search scheme in his Bachelor thesis.

Last but not least, I thank my parents and Laura for their support during this busy time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online Social Networks (OSNs), such as Facebook or Google+, are network-based communication systems that allow their users to create persistent digital representations of themselves, called user profiles, and to establish explicit connections to other user profiles within a bounded system. Those connections, called friendships, are assumed to reflect acquaintances among users of OSNs. State-of-the-art OSNs (in 2015) encompass plenty of communication and sharing functionalities. The range of communication features is spanning from asynchronous one-to-one messaging to sophisticated event organization tools. Media files, such as photos or videos can also be shared with friends or strangers.

OSNs have become popular services that attract more than one billion users[1]. The ubiquity of mobile devices, which support accessing OSNs from anywhere, causes OSNs to be powerful communication tools that change the way how people interact with each other. OSNs gain importance for the daily life of their users by becoming an access channel to parts of their social environment. The social nature of the service facilitates users to share plenty of personal information on OSN platforms.

However, by sharing personal data in OSNs, users become content publishers. It puts OSN users at the risk of suffering undesired side-effects. We denote those undesired side-effects adverse effects in this thesis. The spectrum of adverse effects that can be found in the literature encompasses granting content use licenses to OSN providers[2], cyber bullying [Campbell, 2005, Zych et al., 2015], having too many party guests[3], facilitating social engineering attacks [Huber et al., 2009], undesired effects in job interviews (human resource managers may use OSNs to investigate the personality of applicants [Rosenblum, 2007, McDonald and Thompson, 2015]), loosing the job due to improper public comments and even imprisoning in autocratic countries[4]. The power of OSNs is thus accompanied with the responsibility for content publishers to decide which bit of information shall be shared with whom.

[1] http://allfacebook.de/userdata/, accessed on 2015-03-06

[2] https://www.facebook.com/legal/terms?, accessed on 2015-07-06

[3] http://www.stern.de/digital/online/facebook-fans-stuermen-geburtstagsparty-im-vorgarten-von-thessa-1692209.html, accessed on 2015-03-06

[4] http://www.firstpost.com/politics/goa-facebook-user-faces-jail-for-anti-modi-holocaust-comment-1538499.html, accessed on 2015-07-17

OSNs provide access control mechanisms to adjust the level of accessibility of information and content items. However, despite their importance for reputation and integrity of individuals, privacy controls commonly are very difficult to use and understand. Multiple studies showed users to be unable to handle privacy controls to meet their sharing needs [Liu et al., 2011b, Krishnamurthy and Wills, 2008]. As a result, many bits of information are shared with too many recipients (over sharing).

Fang et al. [Fang et al., 2010] proposed to decrease the frequency of explicit acts of audience selection by applying methods from machine learning to pre-configure the privacy settings. Lipford et al. [Lipford et al., 2008] proposed to check the correctness of the chosen settings by allowing the users to view their profiles as their audience would do and Egelman et al. [Egelman et al., 2011] suggested to employ Venn diagrams to determine sets of recipients. However, users still need to understand and handle the existing privacy controls. To that end, we propose a new interface, that simplifies the audience selection by applying a color coding. In contrast to the related work, the new interface is based on a mental model that is similar to well-known traffic lights. It thus reduces the cognitive overhead when selecting the audience. Both the errors and the effort in selecting the audience are drastically reduced when the new interface is used instead of Facebook's original privacy controls.

Beside adverse effects which potentially occur in case of sharing bits of information with the wrong or too many recipients, users of today's popular OSNs are subject of ownership implications: The market dominating OSNs are each owned by a single commercial company which is the explicit authority in the network. We denote these explicit authorities OSN providers. Users need to trust the latter not to misuse the power, accompanied with being the operator of the system, as well as to be able to protect the system against attackers both from outside and from inside the provider's organization itself. Economic pressure to earn money due to provider-side infrastructure and maintenance costs and the provider's legitimate profit interests lead to strong incentives for OSN providers to monetize user data far beyond the user's content sharing interests [Falch et al., 2009].

Avoiding those ownership implications is the scope of decentralized OSNs (DOSNs). The common ground of multifarious DOSN approaches is that they need to find a way to satisfy the demand for technical resources to realize OSN functionalities, without relying on technical infrastructure that is owned by an explicit network-wide authority (i.e. OSN provider). For example, maintaining user profiles requires storage resources. In addition, messaging as well as content sharing requires network bandwidth. DOSNs thus either apply P2P technology, enabling direct communication amongst OSN users, or introduce federation protocols to realize decentralized client-server architectures (e.g. RFC-822). Also, a combination of both, federation and P2P, has been suggested [Paul et al., 2014a].

In general, DOSNs approaches suffer from the necessity to integrate decentralized resources to realize OSN functionalities. In addition, P2P-based DOSN suffer from the fact that nodes that provide resources are individually unreliable. Resources thus need to be efficiently replicated to provide a reliable service. These DOSN challenges are addressed by complex sets of communication protocols that embody different DOSN architectures. However, compared with today's popular OSN such as Facebook or

Google+, DOSNs still lack functionalities and performance. Even paramount features, such as finding user handles with privacy, are missing.

Spurred by the ambition to mitigate the side effects, resulting from the existence of omnipotent OSN providers, we endeavor to contribute DOSNs to become widely adopted. We contribute a search scheme for finding user handles with privacy, suggest a storage mechanism to help P2P-based DOSNs to be applied under challenging environments and propose to prefetch videos, based on social relations, to improve the performance of DOSNs.

1.1 Problem Statement and Research Questions

The problem, which is addressed in this thesis, is to mitigate adverse effects of using OSNs. Derived from this problem, we work on the following research questions:

1. As a first step, we need to understand how users use OSNs and quantify behavior and threats. To that end, we perform a measurement campaign that sheds light on: How intensive do users use OSNs? How do users orchestrate (combine) the multifarious functionalities within OSNs? Which information do users share with whom in OSNs?
2. The measurement campaign highlights the usability of privacy controls to be one major source of adverse effects. We thus ask: How can we simplify the audience selection to avoid errors when setting the visibility of content? How can we minimize the necessary effort in operating privacy controls? The answer to those questions is the new color-based privacy control interface C4PS.
3. Regardless of privacy controls, OSN users are still at the system operator's mercy not to misuse their data. To that end, we contribute to improve DOSN by answering the questions: What is the state-of-the-art in the field of DOSNs? How we can improve DOSNs? What are the remaining challenges of decentralizing OSNs? We thus survey the state-of-the-art in Chapter 4.
4. We further detail the the question of how we can improve DOSNs by asking the following subquestions:
   a) How can we find user handles without leakage of private information? We present a privacy preserving search scheme in Section 5 without leveraging a search index that encompass user-linkable information.
   b) How can user profile availability be improved in P2P-based OSNs? To answer this question, we present Lilliput in Section 6.
   c) How can we avoid delays in DOSNs to improve their user experience? Video prefetching can reduce startup delays. Prefetching requires predictions on which content will be consumed in the near future. We examine the feasibility of predicting future video consumption based of locally available data in Section 7.

1.2 Contributions

This thesis presents a diverse set of contributions in different fields (Figure 1.2), each of them aims to help users to avoid side effects of OSN usage by keeping control over their data.

This set of contributions can be divided into two major groups: The first group of contributions consists of exploratory contributions (denoted Ex). They shed light on how users use OSNs and identify potential technical improvements of DOSNs. To that end, we conduct user studies and survey the technological state-of-the-art of alternative OSN architectures (DOSNs). The second group of contributions (denoted Tx) technically improves DOSN by proposing building blocks that target remaining challenges in the field of DOSN.

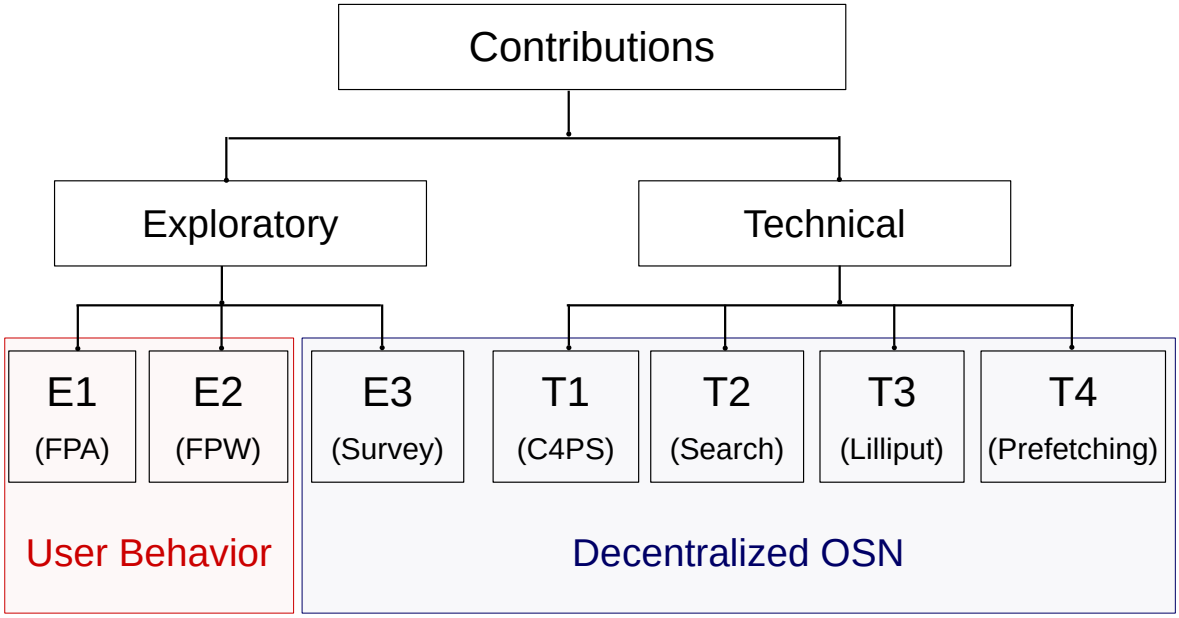


Figure 1.1: Thesis contributions

The following exploratory contributions help to understand the user behavior and the research area of DOSN:

E1: We argue that it is crucial to understand OSN user behavior to estimate the scale and the exact origin of the occurring adverse effects to avoid the latter. Previous studies in the field of OSN user behavior are either based on crawled public profile data, surveys or observed network traffic, thus suffering the limitations of these data collection methods. We contribute a user study with 2071 participants that is based on client-side data collection. We examine how users orchestrate OSN functionality in today's most popular OSN, Facebook, hence acting in their natural environment on their own data for their own reasons equipped with users' access rights.

E2: With the goal in mind to contribute in building privacy preserving OSNs that fit user's needs as good as possible, we elaborate the user's privacy desiderata.

While other studies ask for preferences, we gather a large dataset directly from users who are actually setting their real settings, on their own, for their own reasons. This dataset contains data from Facebook users originated from a variety of countries. Our results show information sharing desiderata to strongly depend on the user's country of origin. In spite of the explicit support for group management, study participants tend to remove fine-grained access rules for their bits of information. They tend to either publish a bit amongst the complete set of friends or to hide it completely.

E3: We present the most widespread survey [Paul et al., 2014a] on decentralized OSNs that examines the state-of-the-art in the field of decentralized OSNs. In this survey, we discuss (D)OSN definitions, identify a minimum set of functionality which needs to be supported by a system to be an OSN and classify the state-of-the-art with respect to the type of decentralized storage, the access control mechanisms and the interaction and signaling mechanisms. We further present a publication timeline to illuminate the temporal development of the field and highlight the uniqueness of each considered DOSN approach in an extra paragraph.

We further provide an evaluation of DOSN weaknesses compared with state-of-the-art centralized OSNs to identify the challenges in the field of DOSNs to tackle for becoming a realistic choice for users. For the sake of completeness, we discuss related approaches that do not constitute a new type of DOSN architecture but aim to improve a certain aspect such as encryption or user profile availability.

The exploratory contributions are the basis for four technical contributions. These technical contributions aim to improve the usability, performance and functionality of DOSNs without negative effects on user's privacy.

T1: Research on the usability of privacy controls figured out that many users are unable to perform the audience selection process for their content in OSNs. Thus, a mechanism that simplifies the audience selection has immanent impact on the privacy. We developed C4PS, a new interface that simplifies the audience selection process to avoid mistakes and to decrease the effort of selecting the audience.

T2: As an insight of surveying the state-of-the-art in the field of DOSNs, we realized that the existing approaches either do not discuss search functionality or assume the search index to be public by leveraging standard DHTs to find user handles. Since an important argument for DOSNs is privacy protection, we decided to develop a search scheme that allows to find user handles with privacy. Our scheme allows to find user handles in systems of decentralized client-server architectures, without disclosing any data that is linked to the permanent user handle (ID or address). The novelty is to avoid building a search index that contains user-linkable data entries.

T3: DOSNs that are based on P2P technology leverage non-reliable resources to store user content. Reliably storing data on unreliable resources requires data replication. In the related work, this is done either by storing and replicating the user data in a DHT or by statically replicating the data within a replication group e.g. at friend's nodes. Authors of approaches that rely on static group

replication mechanisms [Koll et al., 2013, Shahriar et al., 2013] assume at least a small subset of nodes to exhibit long online durations. In contrast, DHT-based approaches encounter difficulties to incentivise resource provision.

We present Lilliput that combines the advantages of friend-based and DHT-based replication. Its dynamic group replication facilitates tit-for-tat cooperation by keeping the number of interacting nodes small. Moreover, it reduces resource consumption to mitigate free-riding incentives by avoiding stale content to be stored and by naturally supporting re-joining of nodes to the set of replication nodes in case of churn.

T4: Since OSNs are widely used to share user-generated videos and DOSNs suffer performance disadvantages as a result of the lack of an information mediator with global knowledge, we examined video prefetching strategies, based on locally-available information. Metadata, such as likes or comments, is technically attached or linked to the respective user-generated media items. We discovered evidence that this information is no feasible fundament for precise predictions of future content consumption and identified two alternative approaches that allow to predict a subset of the content to be later consumed. The first approach is to prefetch content that is authored by a small subset of close friends. The second approach is to evaluate the pre-click time. Users tend to spend more time for scanning those content items that they will consume later. User attention can thus be leveraged to initiate the download process to reduce the video startup delay.

1.3 Thesis Structure

This thesis is structured by eight chapters. Subsequent to this introductory chapter, we present two user behavior studies in chapters 2 and 3 that explore the problem space, followed by contributions to improve DOSNs.

Since the general topic of avoiding adverse effects of OSN usage is very broad, including a comprehensive related work chapter that covers every contribution seems to be impossible. Thus, every chapter contains a related work section that integrates the respective contributions into the related work. Each chapter also encompasses a summary. Below, we depict the content of each chapter in the remainder of this thesis at a glance and highlight the connections amongst these chapters.

- **Chapter 2: Feature Orchestration and Service Usage in Facebook**

  Mitigating adverse effects of using OSNs requires an understanding of how users use OSNs. The remainder of this thesis hence starts by evaluating how users orchestrate the multifarious OSN functionality in the most popular OSN, Facebook, and the content sharing and consumption habits. We elaborate who publishes which types of content and who communicates with whom, how and how often. The results of this chapter are published in [Paul et al., 2015b].

- **Chapter 3: Privacy Preferences of Facebook Users**

  In this chapter, we elaborate the privacy preferences of Facebook users and the actual exposure of their data. Since users are commonly unable to correctly

choose their preferred audience, evaluating actual privacy settings in Facebook does not necessarily reflect user's content sharing preferences. To be able to study content sharing preferences on Facebook in spite of its incomprehensible audience selection mechanism, we introduce and publish a new interface in the shape of a browser extension. It simplifies audience selection and reduces oversharing of content which is caused by audience selection errors. Hence, the browser extension is both: a tool to study privacy preferences and a first solution to mitigate adverse effects. The chapter is based on [Paul et al., 2012c] and [Paul et al., 2015a].

- **Chapter 4: Improving Privacy by Decentralizing OSN**

  DOSNs abolish OSN providers together with their implications and adverse effects. Nevertheless, DOSNs have not been widely adopted so far. In this chapter, we survey their state-of-the-art and classify the surveyed solutions. We identify remaining challenges in the field of DOSN. The chapter is based on [Paul et al., 2014a].

- **Chapter 5: Finding User Handles with Privacy**

  OSNs allow users to find other users' profiles. The OSN provider is the mediator who makes sure that the search mechanism protects users from mass data collection while allowing legitimate requests to be performed. DOSN approaches avoid introducing mediators that need to be trusted. They either rely on out-of-band communication to exchange user handles or define information, which is necessary to specify sought users, to be public. The latter case allows to leverage P2P-based search mechanisms to find user handles.

  However, assuming data that allows to identify users to be public is in conflict with the design goal of DOSNs to protect user data. To this end, we develop a search scheme that allows to find user handles with privacy. The results of this chapter are published in [Paul et al., 2014b].

- **Chapter 6: Increasing Profile Availability in P2P-OSN**

  We also identified the availability of user profiles in P2P-based DOSNs to be an open issue while surveying the state-of-the-art. Furthermore, we use the findings of the user behavior study in Chapter 2 to envision a lightweight DOSN. We argue that it is beneficial to store only the latest updates instead of the whole history of news in the user profiles.

- **Chapter 7: Leveraging Locally-available Data to Apply Video Prefetching**

  Collecting information from several different sources causes delays in DOSN and sharing user-generated content, such as photos and videos, is an important functionality of OSNs. Furthermore, there is a trend to access OSNs from mobile devices. This mixture of different facts motivate us to investigate the feasibility to avoid delays, caused by both the architecture of DOSN and the downsides of cellular networks by prefetching videos based on locally available information.

  In Chapter 7, we present a twofold user study that shows popularity measures, such as comments and likes in Facebook, not to be a feasible basis to predict future video consumption. In contrast, close social relations are strong moti-

vators for users to watch videos. The results of this chapter are published in [Paul et al., 2015c].

- **Chapter 8: Summary and Conclusion**

  In this chapter, we first summarize the thesis, draw conclusions from our work and depict future research directions.

1.4 Work and Collaboration

This thesis is an original work that I did by myself. However, this work would not be the same without collaborations with my supervisor, Professor Strufe, my colleagues in Darmstadt and Dresden, the co-authors of my publications and the students with whom I worked together. Beside the tradition in science to use active voice in plural even in case of a single author, these collaborations are an additional strong reason to use the plural term (we) instead of the singular in this thesis. In contrast, this section explicates the role of myself as well as the input from my collaborators for the contributions in this thesis. It is the only section in the main body of this thesis, where I use a first-person narrator in singular.

Chapters 2 and 3 are collaborative works with Daniel Puscher who wrote his Bachelor thesis and his Master thesis under my supervision. During the work at his Bachelor thesis, he built a prototypical realization of the idea of the color-coding based audience selection interface which was suggested by Professor Strufe. After finishing the Bachelor thesis, he kept working with us as a student helper and developed a browser extension that implements the interface. The second browser extension that allows to study user behavior in detail as well as the R-scripts to evaluate the collected data were part of his Master thesis of Daniel Puscher which he did under my supervision.

The "C4PS" study (Section 3.1) was also supported by Matin Stopcynski and Melanie Volkamer who helped to extend the study, previously conducted in the Bachelor thesis of Daniel Puscher. Martin also helped to improve parts of the publication, acquired participants for the study and helped in data evaluation and visualization.

The DOSN survey in Chapter 4 was a collaborative work with Antonino Famulari, who presented the first version of the classification. The final version of the classification as well as the final text of the respective publication are my own work, guided by Professor Strufe. The visualization of the publication timeline was done by the student helper Stephen Stephen under by supervision.

Lilliput was my own idea. However, Niklas Lochschmidt developed the churn generator, the communication protocols and the simulation model under my supervision in his master thesis. The publication was realized with the help of Professor Anwitaman Datta from NTU in Singapore, who suggested the split design of Lilliput during our discussions and contributed text improvements.

The Search scheme, including the algorithms, also was my own idea. The implementation in Tigase as well as the evaluations were done by Marius Hornung during works for his Bahelor thesis under my supervision.

Leveraging the data from our user studies to improve prefetching of videos in OSNs is a result of discussions with Stefan Wilk. He supervised a student's Bachelor the-

sis work with the goal to improve video prefetching. Stefan's study with 34 participants was conducted by creating an application for mobile phones. It was too small to be meaningful but a perfect supplement for our data which was collected by using a browser extension for Firefox and Chrome. We combined our results and published them together at the CCNC thereafter.

Chapter 2

Feature Orchestration and Service Usage in Facebook

Social networking is a fascinating phenomena. It supports basic human needs such as communication, socializing with others and reputation building. However, the usage of OSNs is still not deeply understood and avoiding adverse effects by developing better OSNs requires an understanding of how users interact with the systems. To this end, we present a study to understand how users orchestrate Facebook's functions to gain benefit for themselves. We also take the changes of user behavior from 2009 till 2014 into account to understand the success and aging process of Facebook, and compare our findings with user behavior assumptions in the literature.

Our study is based on data, which is collected at the client-side. We gathered it from 2,071 users via a web-browser plug-in to overcome limitations of crawled datasets ([Catanese et al., 2011, Meo et al., 2014, Jiang et al., 2013, Gyarmati and Trinh, 2010]), click streams [Schneider et al., 2009] or social network aggregator data [Benvenuto et al., 2009]. Our plug-in is able to measure client-side activity such as scrolling or deactivating tabs to estimate the time that users invest to examine newsfeed posts. The plug-in has access to profile details endowed with user's rights and is able to read activity logs that encompass historical actions regardless of their origin from mobile or stationary devices.

To find volunteers for this study who are eager to install our plug-in, we sent a solicitation to join this study to users of our previous work where we created a new interface to simplify audience selection. In the previous work, we published a browser extension (plug-in) for Firefox and Chrome, called Facebook Privacy Watcher (FPW), which implements this new type of interface with the purpose to yield benefit to people. Both versions (Chrome and Firefox) of the FPW together were installed by more than 44,000 Facebook users.

We asked the FPW users to join a user study about user behavior in Facebook by installing a second browser extension that anonymously collects the data for this study. 2,071 FPW users allowed us to evaluate their user behavior in detail by installing a second browser extension, called Facebook Privacy Analyzer (FPA). We collected basic

demographical data such as gender and age, data on usage patterns with respect to functionalities, data about communication partners w.r.t. to the social graph distance as well as metadata about the shared content. This metadata consists of the type of content, the time when it was created or watched as well as its size in bytes (if available). We respect the privacy of all probands by not storing or evaluating any content or identifier!

To understand the user behavior on Facebook, we evaluate the dataset with focus on the questions: How do people orchestrate the vast variety of functions? Who produces content in Facebook? How much and which kind of information do people share on Facebook? Who consumes which content? How old is the shared content until it is viewed and how long is it commonly consumed? How does the observed user behavior change over time?

The main findings are that Facebook sessions are very short, compared with assumptions in the literature and users' content contributions are extremely disparate in type and quantity. A major share of newsfeed stories is posted by a minority of users and consists of reshared, liked or commented issues rather than original user-generated content. Facebook manages to compensate this lack of high quality content by transforming commercial posts into regular newsfeed content that is accepted by FPA users equally beside user-generated content.

With this work, we contribute at shedding light on the usage of Facebook with respect to churn, content contribution and consumption, as well as communication patterns. We also help developers of alternative (e.g. P2P-based) OSN architectures to make well-founded design choices. Moreover, we provide evidence for Facebook to be an aging network by analyzing dynamics of user behavior over time. Evaluating the history of user actions from 2009 till 2014, we show that users tend to befriend with less other users. Also, actions that cause little effort and commitment, such as reshares and likes, recently became more popular than status updates and comments.

The remainder of the chapter is structured as follows: We first describe the experimental setup in Section 2.1. Thereafter, we examine the attention that users pay to Facebook by evaluating the session durations and frequencies (churn) in Section 2.2 and evaluate the popularity of different functionalities in Facebook in Section 2.3. Furthermore, we elaborate the newsfeed with respect to content creation, composition and consumption in Section 2.4, evaluate communication patterns of FPA users in Section 2.5 and examine dynamics in user behavior in Section 2.6. In Section 2.8, we summarize our work and draw major conclusions.

2.1 Experimental Setup

In this section, we describe our ethical considerations regarding this study, the data collection process, the amount and the composition of the data that we collected. We further describe the bias of the data with respect to the differences to the complete set of Facebook users.

2.1.1 Ethical Considerations

We acquired our participants by asking FPW users to participate in this study. Before installing the FPA, we explained the reason for collecting the data to our study participants and allowed users to access and verify all data before sending it to our server with consent. We further did not violate any rule on Facebook since we directly gathered our data from our participant's browser.

All data that we used for this study is anonymized and encrypted for transmission with state-of-the-art technology. We did not collect or store any content or messages but metadata about the user behavior such as content types, time stamps and hashes of ids to be able to distinguish amongst actors without being able to identify individuals. Nevertheless, we keep the collected data confidential to protect all study participants from deanonymization attempts and do only publish aggregated data.

2.1.2 Sample Generation

We published the browser extension (FPW) to simplify audience selection for posts on Facebook. Enclosed into an FPW update, we asked our users whether they would be willing to help us with a user behavior study and thus share the necessary data with us. 11,572 FPW users filled out the questionnaire: 11.8% of the users answered "yes" 21.3% answered with "maybe" and the rest answered with "no". This questionnaire was intended to be a risk reducing pre-test before developing the data collection plug-in to make sure that a reasonable number users are willing to help us.

Subsequently to this successful pretest, the development of the data collection plug-in started as a statistic module of the Facebook Privacy Watcher and was thus called Facebook Privacy Analyzer. However, we decided the FPA to be a stand-alone plug-in.

The FPA version for the Chrome browser was published at Chrome web store on 24th of November 2013. After fixing several bugs, the Firefox version was published on 16th of January 2014. FPW users who previously agreed to join the study were immediately asked to install the plug-in by showing a pop-up window in FPW. All other FPW users were asked a second time again two weeks later. To incentivise study participation, the data collecting plug-in FPA provided statistics to users about their own behavior. The statistics also could be shared on Facebook and compared with friends.

2.1.3 Sample Bias

Since we cannot force any randomly chosen person to join our study, we only studied persons who were eager to help us in doing research. The data that we collected is thus neither a result of a random sampling process nor the complete Facebook dataset. We thus suffer from a bias that is evaluated in this section.

The majority of our participants joined us by following an invitation via pop-up message in the FPW. The FPW became popular by newspaper articles and radio station broadcasts that reached also less technology-savvy people of several ages. Beside some

international media such as “tech.tavaana.org” or “Der Standard”, the center of FPW news coverage was in Germany. Hence, the overwhelming majority of the FPA users are originated from Germany (84.87%), too. The rest of the participants are - according to the information in their profiles - from 45 other countries.

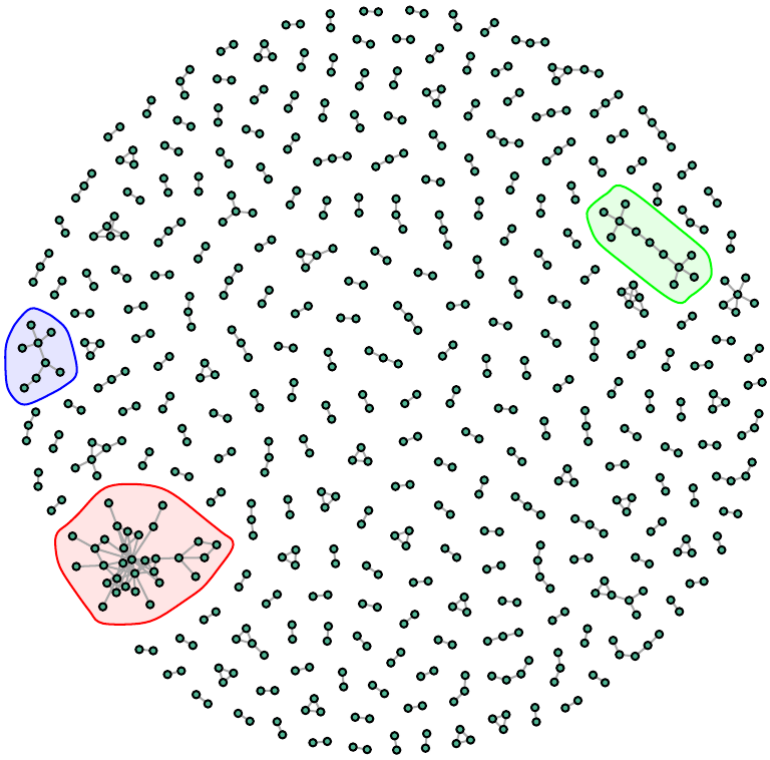


Figure 2.1: Friend relations amongst FPA users; singletons are excluded

The structure of friend relations amongst our study participants is illustrated in Figure 2.1. It includes all (621) FPA users who have at least one friend who is also participating our study. Singletons, which represent 68.02% of the graph, are not part of this illustration to keep it simple. This figure shows that our set of participants is not consisting of a closed community, since only three bigger clusters of 31, eleven and eight users exist. Instead, it is heterogeneous sample.

To further estimate the sample bias, we recorded the gender and the year of birth of our participants from their user profiles. 77.6% of the participants are male, 21.13% are female and the rest did not share this information with us. Some users claimed to be born before 1925 or after 2010. However, we assume the majority of age information to be correct. The majority of our participants was born between 1960 and 1985. Our median participant is a 44 years old male Facebook user. Compared with average Facebook users[1], our participants are older and males are overrepresented.

[1] `http://www.prdaily.com/Main/Articles/The_average_Facebook_user_is_getting_olderand_more_13483.aspx`, accessed on 2015-11-02

2.1.4 Details of Collected Data

To understand the Facebook usage, we require data that describes both components of interaction together with the respective timing information: the actions performed by users as well as the information flow from Facebook to the users. We mainly collected four data types, using the FPA: the performed actions of users, the friend lists of users, activity logs and basic demographic information about the FPA users. Furthermore, we measured exact session durations by storing the time when activating and deactivating browser tabs in case of active Facebook sessions. In the remainder of this section, we explain the four main types of collected information and their necessity for our study.

Performed Actions

The occurrence of every action that a user performed in Facebook was recorded in the (local) database, together with a timestamp and the browser tab ID. We recorded further metadata about the actions such as the hashes of persons who are involved in those actions.

Friend Lists

We recorded the friend list as set of UIN hashes. We needed this information for several reasons: We checked whether two-sided actions such as messaging or profile views are performed amongst friends or strangers, we counted the number of friends of each FPA user to calculate the node degree within the ego-graph and we checked newsfeed posts whether they are originated from friends.

Activity Logs

The FPA can only gather data in case the user uses a web browser based Facebook access. However, Facebook maintains an activity log as a part of the user profiles. This activity log contains activity records back till the time of registration at Facebook independent from the access channel (e.g. mobile app or browser).

These records contain almost all actions which have been performed on Facebook together with the timestamp and some metadata such as communication partners. Private messaging e.g. is not included in the activity logs. Nevertheless, it is a very valuable data source for our analysis since it allows us to estimate the fraction of actions that we can observe in the browser. We can thus bridge the gap that would appear in case of only evaluating data from Firefox or Chrome browsers.

Furthermore, due to the activity log's long term records, we can evaluate changes in user behavior during time. We can thus trace the learning process of new users joining Facebook.

User Demographics

Based on ethical considerations to protect user's privacy, we only stored basic user data such as age and gender. We need this data to estimate the bias of our sample.

2.1.5 Data Quantification

Since we changed parts of the code basis through updates after the first publication of the plug-in, we decided to use only data that was collected after 1st of January 2014. During our observation period of 123 days, 2071 users installed the FPA. However, not every study participant joined at the first day and not every participant stayed the whole rest of the time. We thus observed our participants on average 34 days.

2.1.6 Mobile Device Usage

In 2014, 68% of all Facebook's users accessed the OSN at least once per month using a mobile device[2]. Since the FPA is an extension for the Firefox and Chrome, no data can be collected while using the Facebook application on Android or Apple devices.

However (as explained in 2.1.4), we obtain activity logs from our users that contain all actions that are performed in the respective account. This also encloses actions on mobile devices or stationary devices without an installed FPA instance. One downside of evaluating activity logs is that we cannot precisely distinguish between actions, perfomed on stationary and mobile usage. We can only estimate the mobile device usage by comparing the FPA included records with those from the activity logs.

In addition to the activity logs, the FPA can read the whole user profile, independent of the devices that have been used to create the content. The participant only needs to access her profile at least once within a browser with an active FPA instance. We thus assume most of our analyses to be unaffected by mobile device usage even though the data is obtained via browser. Evaluations, that suffer drawbacks due to the gap in mobile device usage, are marked accordingly.

2.2 Churn

Churn denotes one component of our user behavior measurements. It describes session starting and ending patterns. This is important, since churn reflects intensity of users using Facebook and hence is a measure of the total attention Facebook receives. Moreover, realistic churn assumptions are the basis to evaluate P2P-based DOSNs, since only nodes that are connected to the system are able to contribute resources. In this section, we describe the churn behavior that we observed by evaluating the session durations and the average session frequency per day.

Caused by the properties of web-based systems in which communication is triggered by user activity (events), different methods exist to measure churn (Figure 2.2). The related work (e.g. [Schneider et al., 2009]) use inter alia the absence of activity (timeouts) as an indicator for users to leave the system. In contrast to the related work, the browser plug-in FPA has access to more precise information such as whether a tab is activated or not. To ensure comparability, we include four different measurements

[2] http://de.statista.com/statistik/daten/studie/380636/umfrage/prognose-des-anteils-mobiler-nutzer-an-den-facebook-nutzern-in-den-usa/, accessed on 2015-11-01

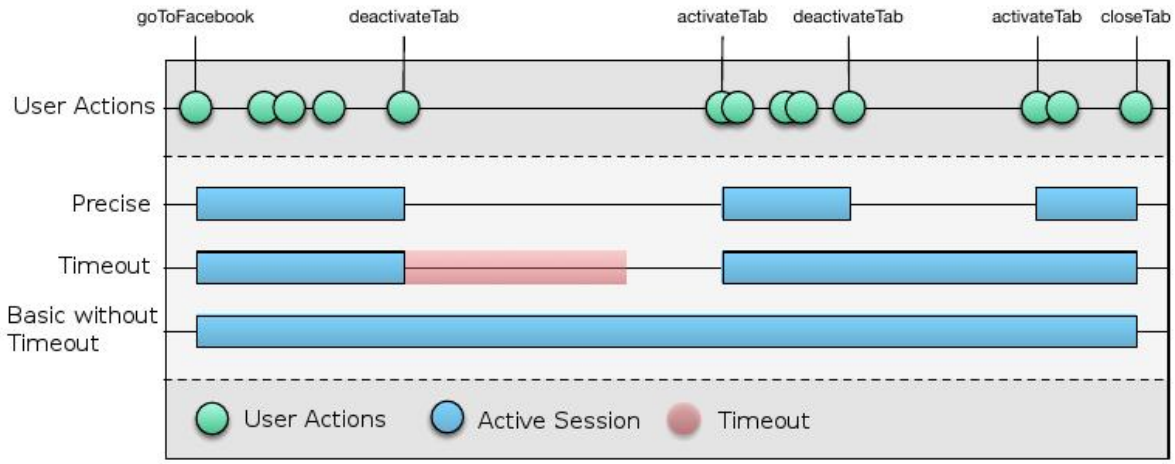


Figure 2.2: Illustration of churn measurement methods; (adapted from [Puscher, 2014])

instead of only presenting the most precise measurements. We distinguish amongst the following four churn measures:

- **Basic:** A session always starts with the login on Facebook and ends either with a logout or with closing the last open browser tab with an open Facebook session. However, this session definition leads to extreme cases of sessions lengths lasting several days. It does not realistically reflect the user's attention. We thus included a timeout of three hours starting after the last action was performed by the user.
- **Timeout:** The timeout measurements are conform to the previous measurements, using a more aggressive timeout of five minutes. We ague that this short timeout reflects user attention better than the basic churn measure [Schneider et al., 2009].
- **Basic without timeout:** We included the previous measurements without timeout to quantify the effect of the timeouts on our basic measurements. Comparing 'basic' and 'basic without timeout' indicates how many users log themselves off or close Facebook tabs while leaving.
- **Precise:** Since the FPA notices tabs to be activated and deactivated, the most precise measurement is to count a session to start as soon as either a users performs a login action on Facebook or a browser tab on Facebook is activated. The session ends in case the browser (or the tab) is either closed or deactivated or a logout action is performed.

Figure 2.3 shows the duration of sessions. The 'precise' and 'timeout' measurements depict very short Facebook sessions of 2:16 minutes on average (median: 0:17) for 'precise' and an average of 5:32 (median 2:21) for 'timeout'. The average results of the 'basic' measurements (31:40 minutes) are roughly in line with the results of Schneider et al. [Schneider et al., 2009] and the measurements without timeout show unrealistic lengths of 240:01 minutes.

Beside the session duration, the average number of sessions is important, too. Multiplying the averages of both measures results in the average total online time per day. Figure 2.4 shows the distribution of average session numbers. The 'precise' measure-

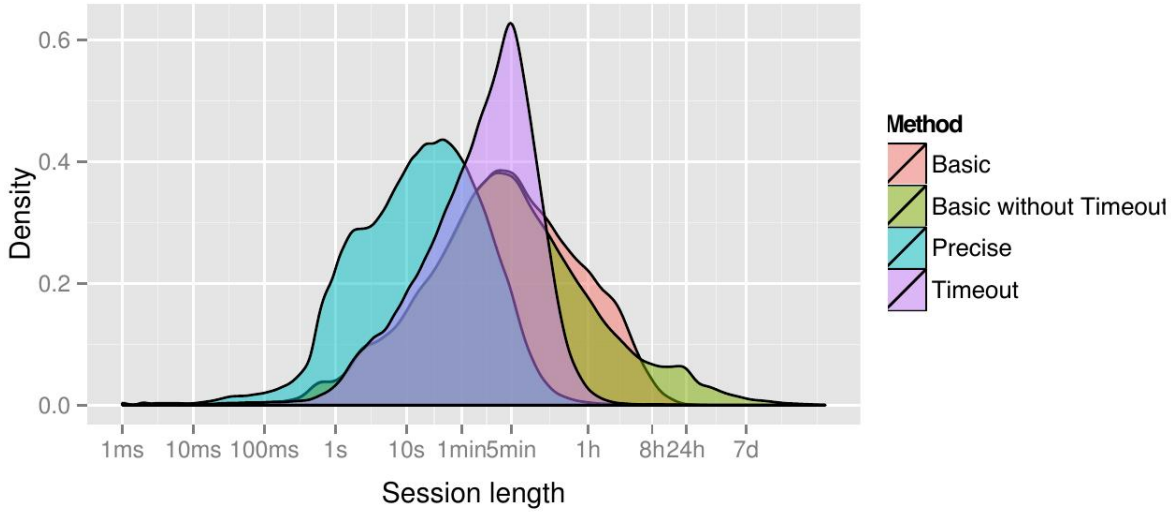


Figure 2.3: Distribution of session durations with respect to four different measurement methods on a logarithmic scale

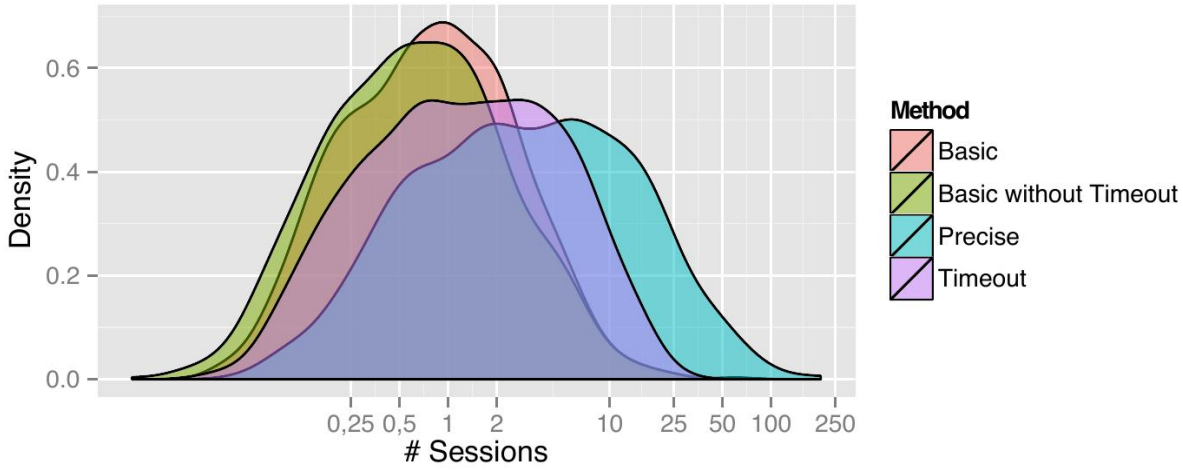


Figure 2.4: Average number of sessions per day

ment indicates an average of 2.97 sessions, the 'timeout' method 1.28 and the 'basic' indicates an average of 0.79 sessions per day.

2.3 Function Popularity

Beside its third-party app ecosystem, Facebook itself comprises a rich compilation of functions which characterize the service. Figure 2.5 contains box-whisker plots which show the relative fraction of time that users spend with each function. The box-whisker plots are ordered by the median time that users spend with each function.

The newsfeed, called 'Timeline', dominates Facebook usage, followed by viewing other users' profiles. Viewing pictures is very popular, too. Surprisingly, users spend more time with topic-related interest groups (named 'groups' in Figure 2.5) and the enclosed newsfeeds ('list newsfeed') than with exchanging messages with others. The median FPA user spends more time with maintaining the own profile than with running

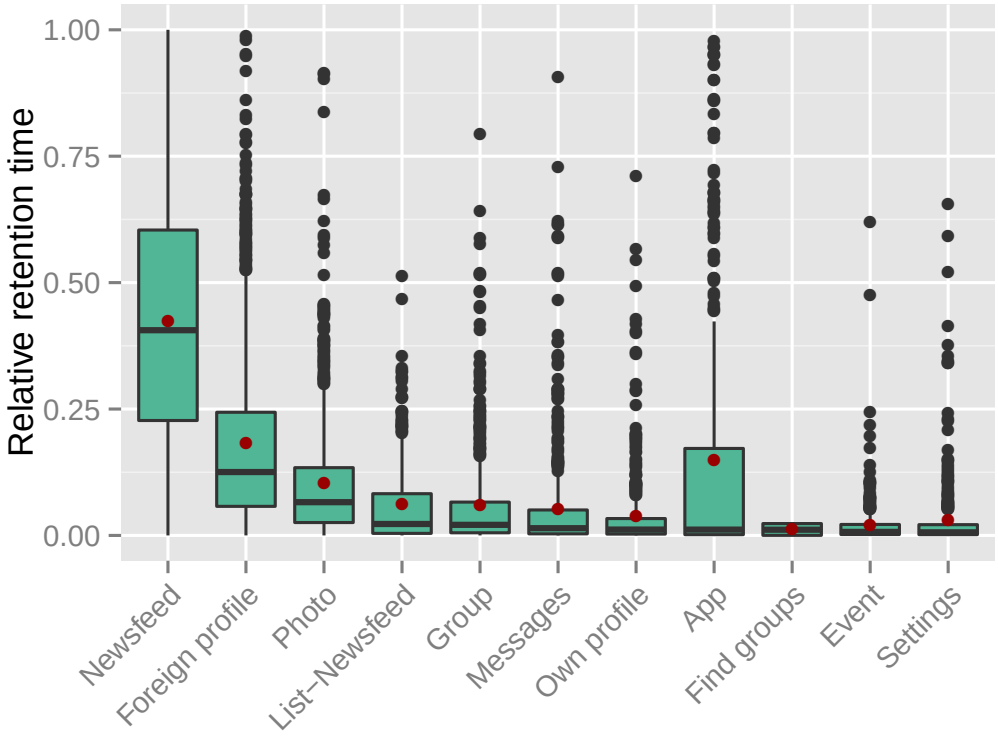


Figure 2.5: Fraction of time that FPA users spent using different Facebook functionalities

apps on Facebook. However, running apps is not unpopular in general. A minority of users spends a big fraction of their time with apps.

The page transition matrix in Figure 2.6 shows relations among different functions in Facebook by depicting page type transitions. Two main findings can be quickly realized: Users tend to repeat actions several times (e.g. view a picture after viewing a picture) and the newsfeed is the dominating functionality getting the most page hits from other transition sources.

Third Party Applications

In this section, we evaluate the usage of third party applications (apps) which are completely integrated into Facebook itself and leverage the Facebook platform to provide benefit to their users. For these evaluations, we only included data from users who never accessed their newsfeed to exclude strangers and those who joined our experiment for less than one week. We thus only used the data of 1,068 users.

The mutual benefit of the app creator and Facebook are that Facebook's functionality is extended by third parties and the third parties can leverage Facebook platform functionality. Based on the platform functionality, the App instances of different users can communicate with each other and the platform allows the apps to receive information about the users, such as the friendship connections or interests, in case of user's consent. The apps can thus leverage the social graph to fortify collaboration amongst friends and to establish or support a feeling of togetherness.

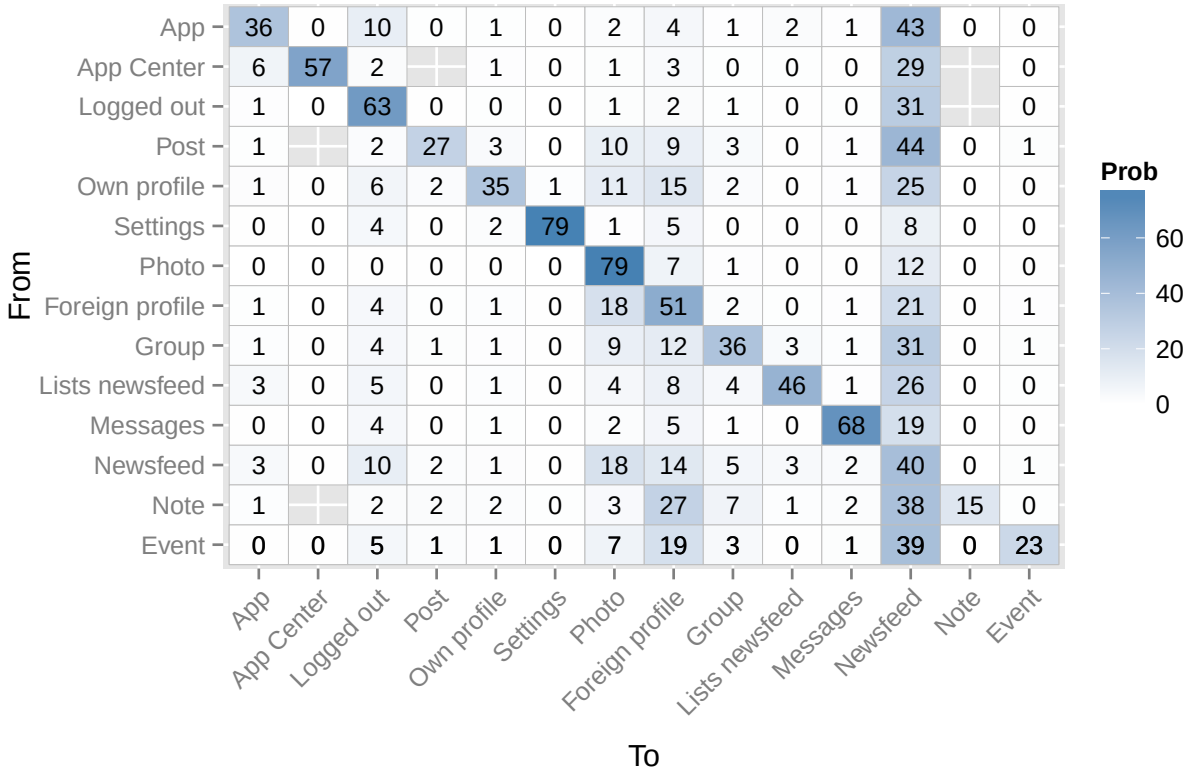


Figure 2.6: Page transition matrix; the row values sum up to 100 %

29.96% of the FPA users used at least one app and 50.88 % of all app users used exactly one app. On average, app users use a total number of 2.55 apps and spend 14.97 % of their time on Facebook with running apps. Figure 2.7 shows the distributions of the number of apps the users from the evaluation set are using as well as the distribution of the fraction of time they spend. Figure 2.8 shows the popularity distribution of apps amongst users. However, because of our privacy limitations, we only know the hash values of app IDs rather than their names. Hence, we only know how popular apps are but do not know which apps are popular.

2.4 The Facebook Newsfeed

Being the core of Facebook, we dedicate this section to the Timeline. Attracting users to generate and contribute content to the Timeline, such as pictures, videos status updates and text messages, is crucial for the success of social networking platforms like Facebook. However, the platform operator needs to solve a chicken-and-egg problem: content contributors need to be incentivised to contribute content by the existence of an audience that is interested in their submissions and the crucial incentive for a potential audience to spend their attention to the platform is the existence of interesting content.

Thus, Facebook's role is to be an information mediator that manages two scarce resources: valuable and interesting content as well as attention of the audience. "The goal of News Feed is to deliver the right content to the right people at the right time so

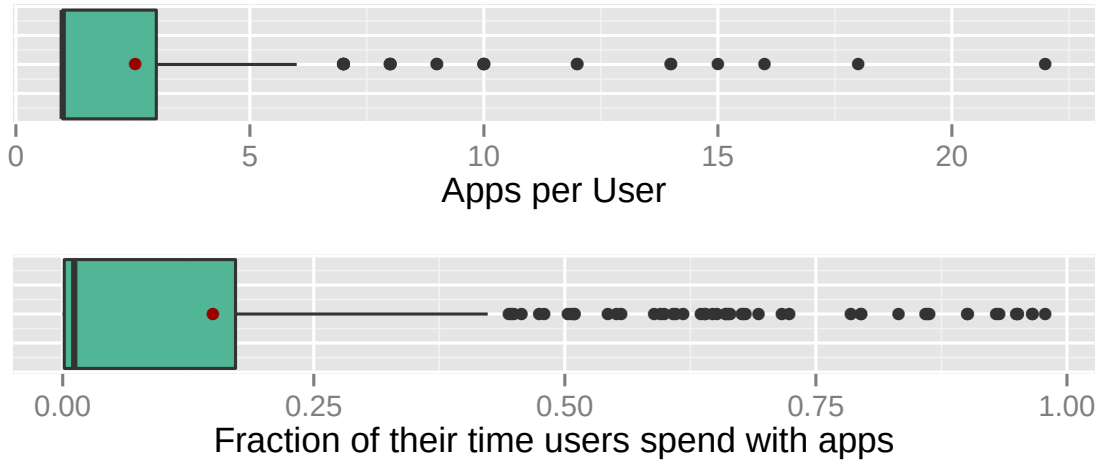


Figure 2.7: Box-whisker-plots: distributions of the number of apps FPA users are using and the time they are spending with apps

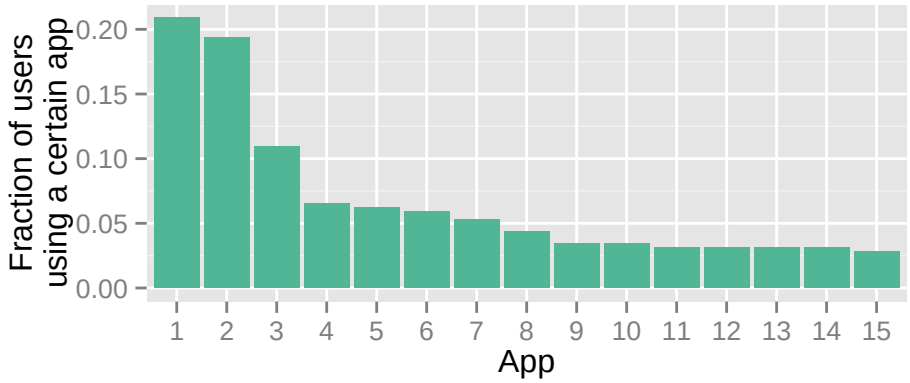


Figure 2.8: Popularity of apps (due to privacy limitations, we do not know the names of the apps and thus numbered them)

they don't miss the stories that are important to them. Ideally, we want News Feed to show all the posts people want to see in the order they want to read them." [3]

The straightforward way to only leverage friendship connections as communication channels while respecting restrictions arising from privacy settings is not sufficient to find an interested audience for content. The matchmaking is a hard task to solve for two reasons: a minority of users posts a lot of content which is not interesting for all of their friends and the amount of posts quickly becomes too high to be read by others[3]. Facebook thus decides which content to place in which user's Timeline to provide the most interesting news to the users during their period of attention.

This matchmaking can be improved by understanding two determinants: the interests of users in content as well as by understanding the meaning of content. In the remainder of this section, we first examine user's content contribution to understand Facebook's initial situation for placing content in timelines. We then explain the news-

[3] `https://www.facebook.com/business/news/News-Feed-FYI-A-Window-Into-News-Feed`, accessed on 2015-10-28

feed arrangement which reflects Facebook's assumptions and wishes which content to view. The consumption of presented content is evaluated thereafter. Finally, we show the impact of the provided content based on the measured user reactions such as comments and likes. The following Timeline related evaluations are based on the subset of 774 users who viewed at least 100 posts (788,938 in total) to avoid outliers to affect our results.

2.4.1 Content Generation

The Timeline contains not only user generated content. Facebook itself is creating a big portion of posts that appear in the Timeline e.g. to inform users about status or profile picture updates of their friends. Companies (ad pages) can also post regular newsfeed messages. However, in the remainder of this section (2.4.1), we focus on user-generated content rather than content generated from Facebook itself or commercial pages.

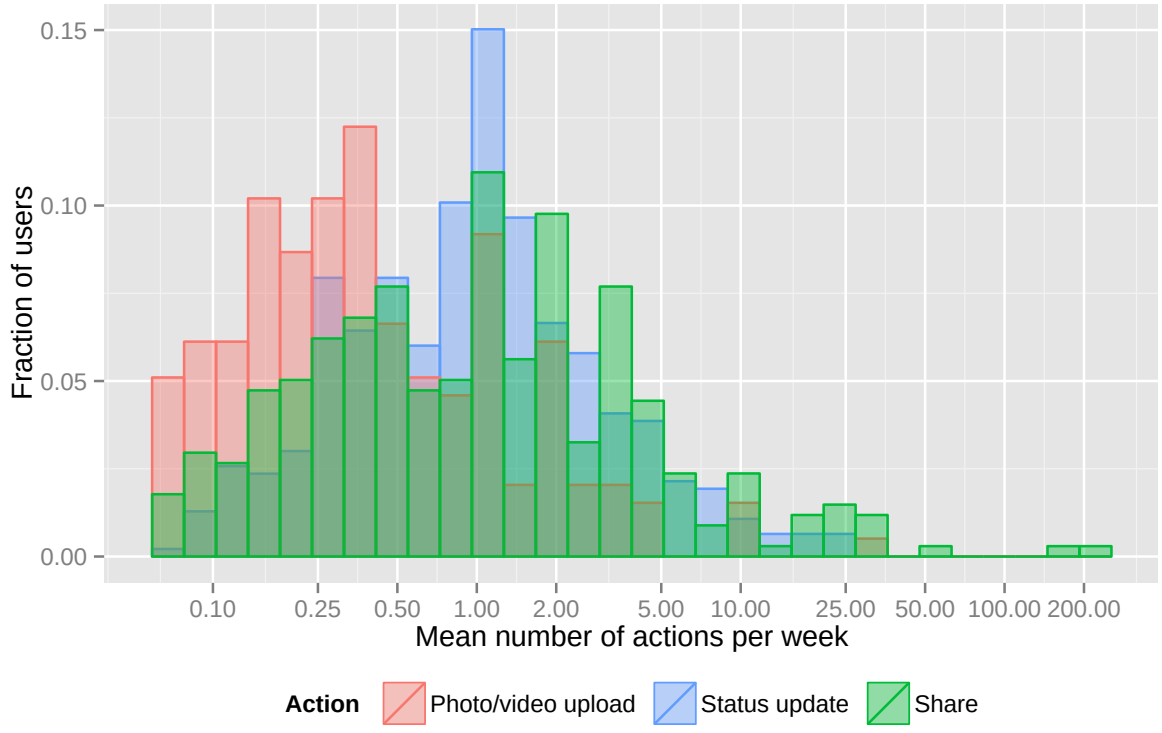


Figure 2.9: Content that was posted at the Facebook Timeline by its type

Users can create different types of content: status updates which only consist of pure text messages, shared links to internal or external pages as well as media such as photos or videos. In general, publishing content is not very popular. 36.37% of the users who viewed at least 100 posts during the observation period did not post anything. We excluded them from further content generation evaluations.

Figure 2.9 shows the popularity of the three kinds of content uploads. The main finding is that the average number of posts per week is extremely diverse with respect to different users. The relative frequency of sharing content can be represented by a

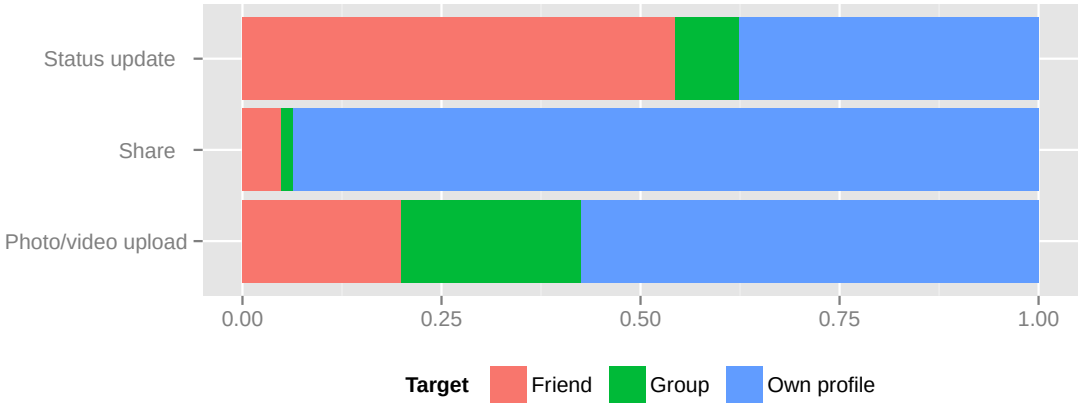


Figure 2.10: Content posting targets

long tailed distribution. For example, we observed users to share more than 200 links per week on average.

However, the overwhelming majority only posts very few items. Our FPA users created on average 4.36 posts per week. Most frequently, they shared already existing content amongst others (2.43 times per week). Status updates and photo and video uploads happen less often. On average, 1.57 Status updates and 0.35 photos were posted.

Beside the own profile, users can also leverage other channels to post content. Figure 2.10 shows what type of content is posted on the own profile, at a friend's profile or on a group's newsfeed. The overwhelming majority of Links are shared on the own Timeline and most photos and videos are published there as well. However, elaborating details regarding status update posts depict different patterns. Facebook invites users to send birthday congratulations via e-mail notification and provides a dedicated page for this purpose. The the result is that the majority of status updates was published on friend's profiles whereof 36.42% of them were directly posted via the sidebar of birthday congratulation pages.

2.4.2 Newsfeed Composition

In this section, we evaluate the composition of the newsfeed. We examine who authors the newsfeed content to depict the nature of the service and we evaluate which fraction of friends contributes to the newsfeed to scrutinize Facebook's eligibility as a tool to keep in touch with friends.

Facebook was initially planned for students to establish and maintain connections among friends. Figure 2.11 shows the distribution of newsfeed entries with respect to the authorship. Nowadays, the average fraction of friend-generated content dropped to 49.5%. While introducing professional pages as a tool for companies to communicate with their customers, Facebook became an important advertising platform that now reaches an average fraction of 41.4% of commercial newsfeed entries. Only a very small fraction of content was initially posted by strangers (9.1%). Content from

strangers may appear in the personal newsfeed in case that friends interact with it (e.g. attach likes or comments).

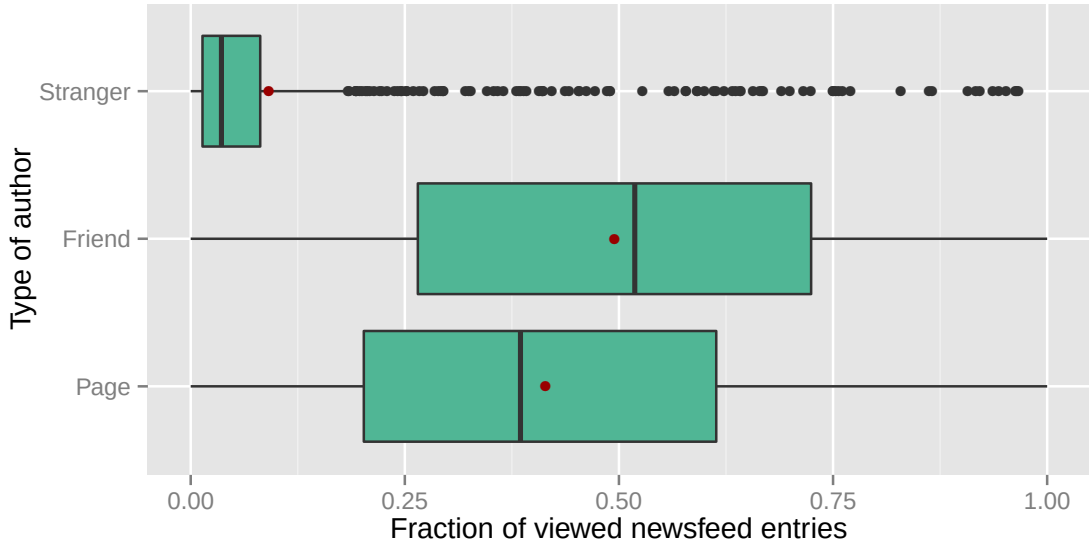


Figure 2.11: Authors of newsfeed entries

The fraction of friends appearing in the newsfeed is critical for Facebook as a service that allows users to stay in touch with the whole set of friends. The intuitive assumption is that the fraction of friends that contributes to the newsfeed of users decreases with an increasing set of friends. This assumption is based on the facts that an increasing set of friends means a bigger set of stories for Facebook to choose for including into the Timeline while users only spend a limited amount of attention to the newsfeed.

Figure 2.12 shows the linear regression on the size of the set of friends versus the fraction of friends in the newsfeed. It shows both: that the decreasing assumption holds as well as that the relation is not very strong (-0.32). Our interpretation is that Facebook tries to include as many friends as possible into the Timeline.

Some users with the huge set of more than 700 friends still see content of more than 50% of their contacts in their newsfeeds. This leads to a huge amount of newsfeed entries and suggests that Facebook scales the number of content items in the newsfeed according to the attention that it receives.

Most newsfeed entries are very fresh until they are shown: 84.79% are not older than 24 hours and 25.77% are created less than one hour before. However, a small fraction of entries is very old: 4.02% are older than 7 days 1.73% are older than 30 days. One reason for very old content to appear or reappear in the newsfeed are friends liking or commenting old content.

2.4.3 Content Consumption

In this section, we provide insights into the content consumption habits of the FPA users. We first evaluate how long different types of newsfeed entries stay in the viewport of the browser before being clicked. This gives an idea about how much effort FPA users invest into the decision which content they view. We then evaluate how many

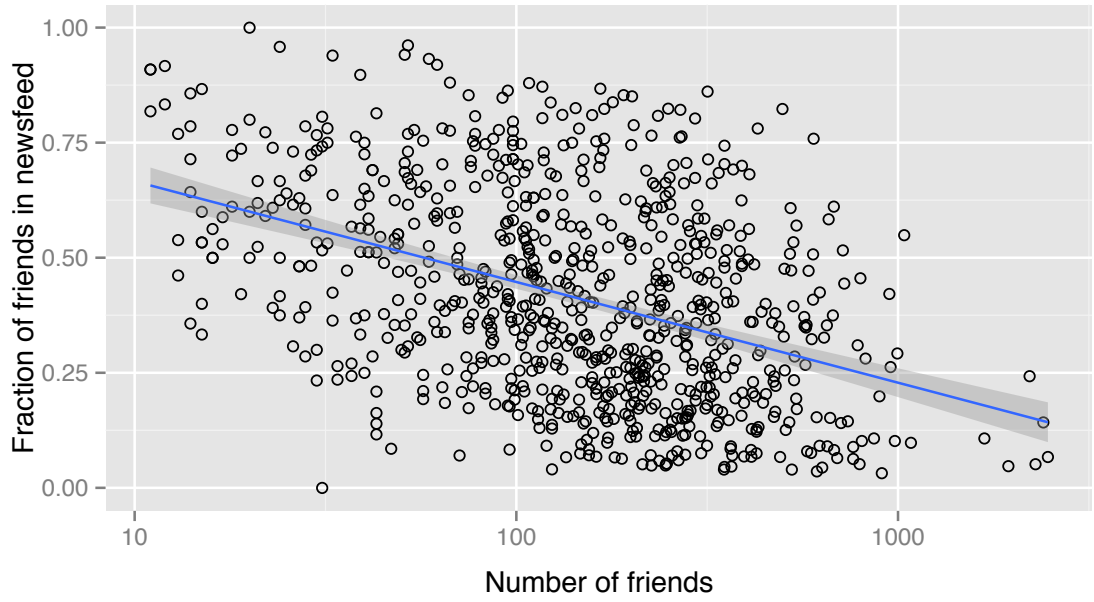


Figure 2.12: Relation between the total number of friends and the fraction of friends appearing in the newsfeed (linear regression)

newsfeed entries were viewed on average per day to estimate the amount of attention a user pays to the newsfeed. Finally, we examine the types of accessed content to allow comparing the posted with the viewed content.

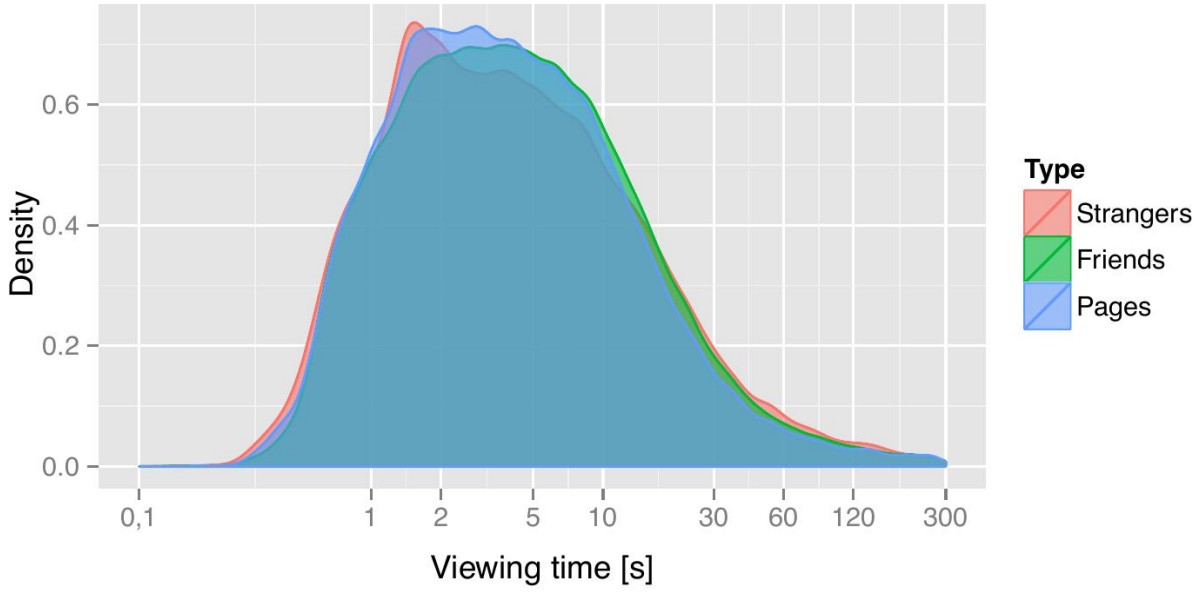


Figure 2.13: The time that newsfeed entries stay in the browser viewport with respect to the authors of entries

Figure 2.13 shows the time a newsfeed entry stays in the browser viewport before being clicked. This information illustrates the time investments of users to check a certain entry. Most users invest between one and ten seconds (average 9.5s) to decide whether to click on an entry or not. This is valid for all types of entries independent

from the authorship. Very interesting is that posts from commercial pages receive a similar attention like posts from friends or strangers (9.8s vs. 9.1s). However, posts from strangers cause a slightly more diverse checking time than others. While there is a peak at 1.5 seconds, there are is also a higher value at 60 seconds compared with posts from friends or pages.

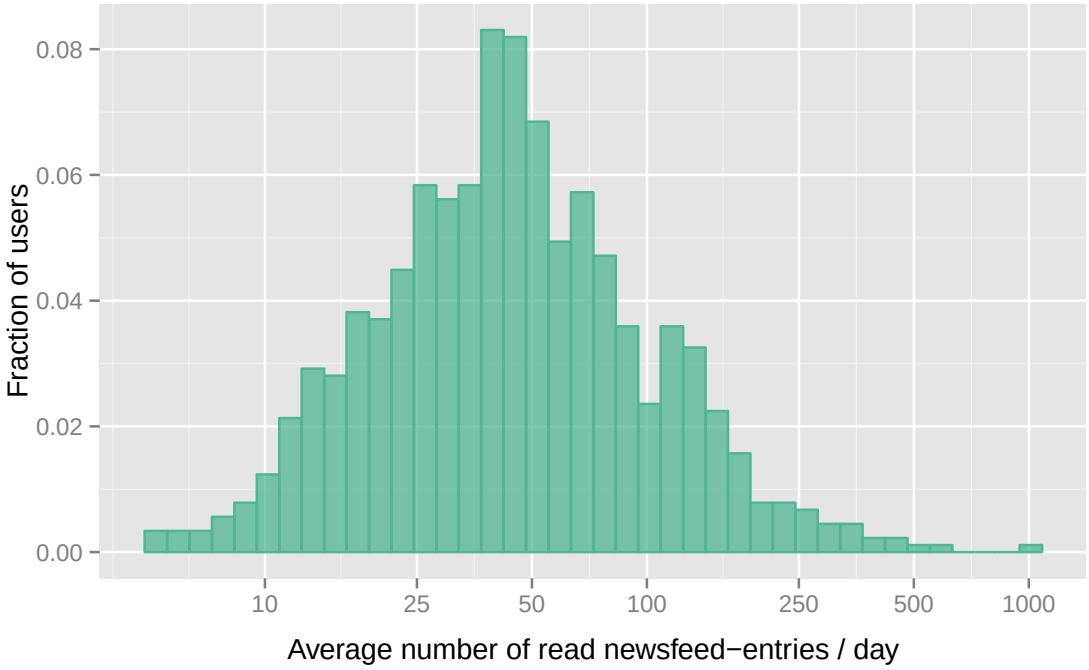


Figure 2.14: Distribution (Histogram) of the average number of newsfeed entries that have been viewed by FPA users during the observation period; each bar indicates the fraction of users viewing the respective average number of posts

On average, users view 43 posts per day. A histogram that allows to estimate the distribution can be found in Figure 2.14. Figure 2.15 shows the composition of newsfeed entry views. The biggest fraction of viewed posts consists of shared links (41.76%), photos (27.34%) and status updates (16.77%). Considering the authorship of posts, it is surprising for us that the clicked shares of commercial posts roughly equal those of friends or strangers. FPA users seem to accept commercial newsfeed posts equally as regular news beside user-generated content.

FPA users like on average roughly 4% and comment 1% of all newsfeed posts. Figure 2.16 shows the long tailed distributions of the number of comments, attached to newsfeed entries.

2.5 Communication Patterns

In this section, we provide insights into communication patterns of FPA users. We first separately evaluate the user profile views as such representing the most popular two-sided communication functionality. In case of viewing user profiles, the profile owners

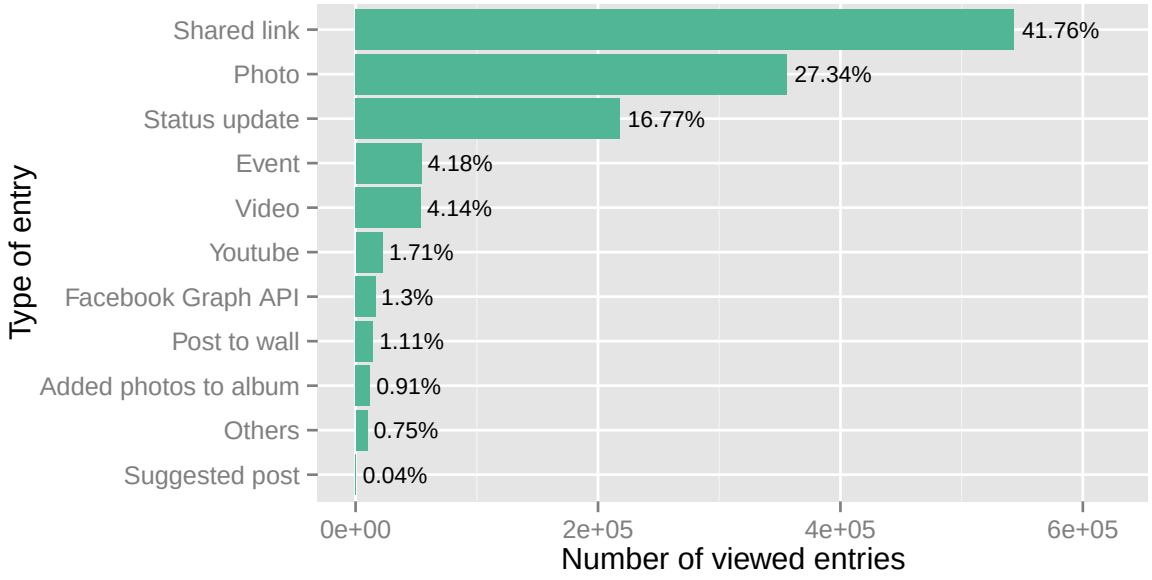


Figure 2.15: Number of newsfeed entries that are viewed by FPA users with respect to entry types

are information sender and the user who is accessing the profile is the information receiver.

2.5.1 User Profile Access

Profiles in Facebook can be classified into two major categories: profiles of individuals and profiles of professional pages that represent companies or prominent persons such as actors or musicians. In Figure 2.17, we further distinguish amongst friend's, friend-of-friend's and stranger's profiles as well as between liked and unliked professional pages. FPA users visit on average 33.51% pages of friends, 30.19 % 'unliked' professional pages 15.32% pages of friend-of-friends, 11.22 % stranger's pages and 9.76 % liked professional pages.

2.5.2 Communication with Friends

Facebook is widely known as a tool to communicate with friends. To understand such communication, we elaborate the two-sided functionalities. Figure 2.18 shows the percentages of friends that have been communicated with during the observation period by using a certain communication function. Since this analysis is affected by a too short observation time, we only included data from 714 users who participated for more than four weeks in our study.

The majority of FPA users communicates with only a minority of friends. By far the most popular communication between friends is viewing newsfeed entries of each other, followed by clicking on shared links, viewing profiles and liking content. On the other side of the spectrum, least popular are poking, timeline posts and commenting. Excluding the extreme cases of poking and viewing newsfeed entries, the most popular functions are those which imply the lowest commitment of the acting user.

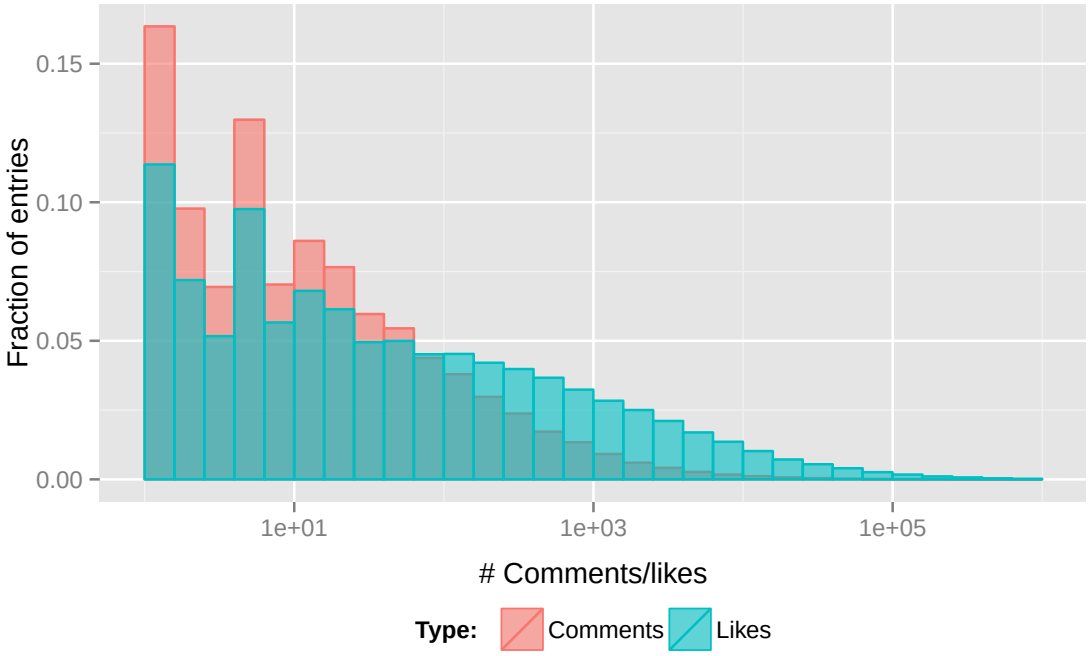


Figure 2.16: Number of comments and likes of newsfeed entries

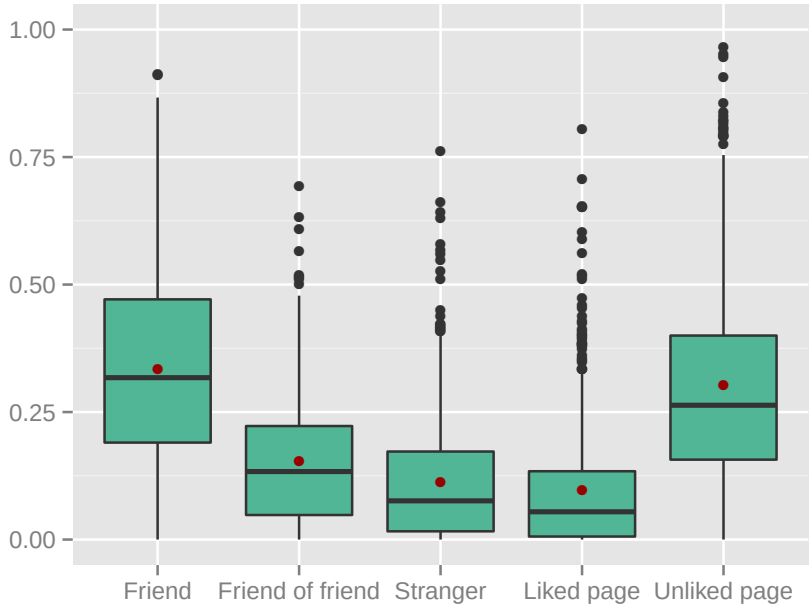


Figure 2.17: Profile page access with respect to the social graph distance; we distinguish between liked and not liked (unliked) professional pages

Both poking and viewing newsfeed entries are exceptional cases for different reasons. The newsfeed is arranged by Facebook's algorithms and thus no explicit choice of the users. Poking is exceptional because it is only intensively used by a very small fraction of users.

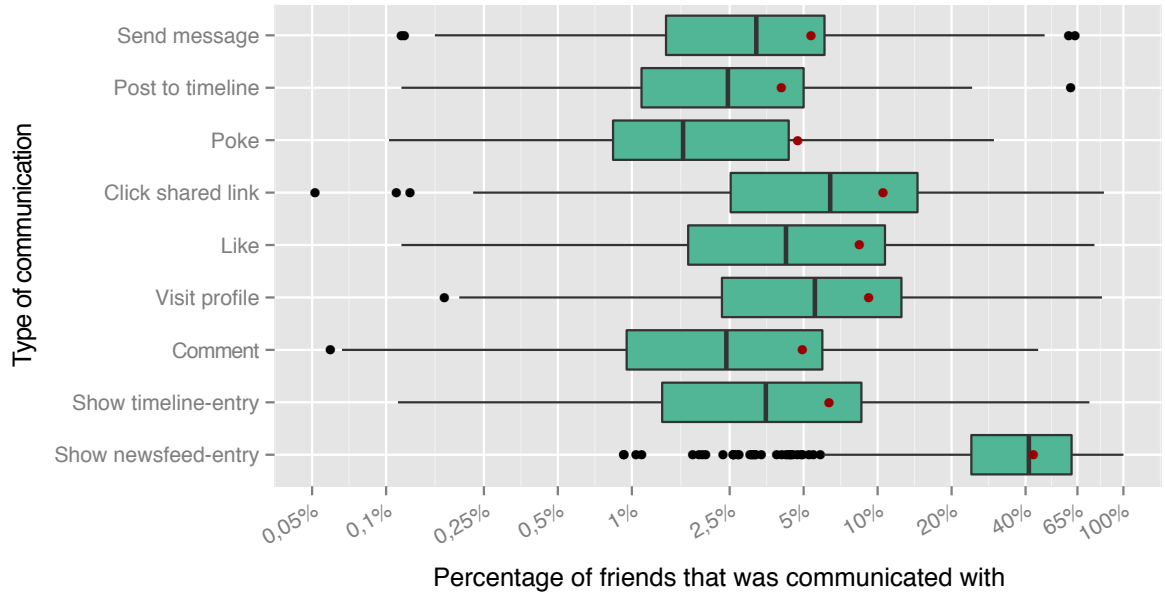


Figure 2.18: Percentage of friends with whom the FPA users communicate with respect to the communication function

2.6 Dynamics in User Behavior

The social networking idea still disseminates amongst world's population. The establishment of best practices as well as the process of users to learn how to use social networking tools like Facebook is an ongoing process. Furthermore, Facebook is permanently working hard to improve the service. Thus, usage patterns evolve over time and examining user behavior in the field of social networking means to examine a quickly moving target.

Fortunately, Facebook encloses an activity log into user profiles. It contains all actions of many categories that users performed, starting from the day when a user registered her account at Facebook. Our tool, FPA, is able to read this information. To show the development of usage patterns, we compare the popularity of the seven most popular activities on Facebook per year in Figure 2.19.

This plot strikingly showcases the maturity of Facebook and its decreased growth rates. The fraction of friend-adding actions dropped from year to year. This indicates that the process of new friends joining the network as well as the establishment of friendship connections converged to the natural social dynamics in the society.

Also noticeable is that the share of actions that require little effort from users increases: Likes and sharing of content recently became much more popular than in 2009. The fraction of sharing actions increased by a factor of 4.62 from 2009 till 2014. Accordingly, the fraction of comment, status update actions and photo uploads decreased.

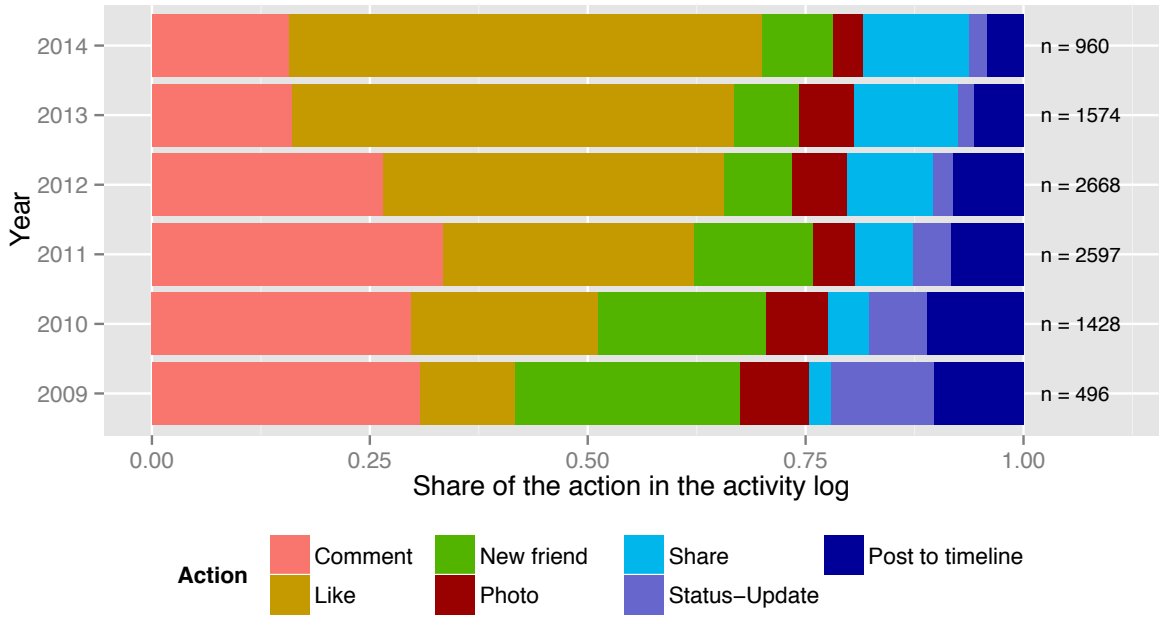


Figure 2.19: Comparison of Facebook usage from 2009 till 2014

2.7 Related Work

A vast amount of related work in the field of user behavior in OSNs has been done by now. Prior work can be classified by the research discipline (e.g. computer science and psychology), the data collection method as well as the social network that has been analyzed (e.g. Twitter, Facebook, Google+).

We aimed to understand the technical core of Facebook's success. In the remainder of this section, we thus focus on related work in computer science about Facebook that evaluates functionality rather than e.g. the social graph. Unlike works in the field of psychology or sociology [Wilson et al., 2012], we did not aim to contribute in exploring OSN user's individual properties or any relation between OSNs and societies. We took the system's point of view and wanted to know how users use Facebook's functionality and how to build excellent systems that support user's needs. Since capturing functionality usage causes stringent requirements on the data collection process, we examined related work based on this criteria in this section.

Many works on user behavior are based on crawler-gathered data [Catanese et al., 2011, Meo et al., 2014, Jiang et al., 2013, Gyarmati and Trinh, 2010]. Datasets that are acquired by employing crawlers contain static elements of user profiles. Dynamics can be estimated by frequently crawling the same information to detect changes. Also activity counters such as page view counters can detect some types of dynamics [Strufe, 2010]. Page view counters have been also used by Lin et al. [Lin et al., 2012]. They crawled Renren and Sina which offer pageview counters and allow to crawl a list of the last nine visitors. This type of information allows insights into profile visits by building directed and weighted graphs. However, datasets that are acquired by employing crawlers are not sufficient for our purpose to understand how users use Facebook since they neither reflect the use of all kinds of functionality

(such as messaging, likes or the 'Timeline' usage) or their interplay nor do they allow to evaluate exact timings.

Schneider et al. [Schneider et al., 2009] analyzed passively monitored click streams of Facebook, MySpace, LinkedIn, Hi5, and StudiVZ. They analyzed feature and service popularity, churn, click sequences and profile usage but do not evaluate information about the newsfeed and profile page compositions or historical data. We can confirm the finding that users are trapped when using a specific functionality. If users view pictures, the likelihood is extremely high that the next action is again to view pictures. However, Schneider et al. did not evaluate content consumption nor content contribution habits of users.

Benvenuto et al. [Benvenuto et al., 2009] elaborated click stream data from 37,024 users of Orkut, MySpace, Hi5, and LinkedIn in a twelve days period, collected by a Brazilian social network aggregator. They additionally analyzed crawler data which was collected at Orkut. The main results of this work are session descriptions containing information such as how long and how often users use which function of the system. With the help of the crawler data on Orkut, Benvenuto et al. analyzed function usage with respect to friend relations. In contrast to this work, we focused on Facebook in 2014. Due to client-side data collection, we know timing issues such as pre-click times, can distinguish between stranger profiles and professional sites and are able to do long-term evaluations based on activity log data.

To overcome these dataset implicated downsides in understanding Facebook usage, Facebook-internal applications and client-side data collection methods have been developed. Mondal et al. [Mondal et al., 2014] leveraged a Facebook app to examine Social Access Control Lists and Luarn et al. [Luarn et al., 2014] developed a Facebook app to test and confirm the hypothesis that people's network degree is positively correlated with the the frequency of information dissemination. Client-side data collection can be found in [Weinreich et al., 2006, Velayathan and Yamada, 2007]. However, these works did not analyze OSNs but web surfing behavior in general.

The allocation of attention amongst friends has been analyzed by Facebook [Backstrom et al., 2011]. The main findings are that Facebook users concentrate their attention on a small fraction of friends while messaging is much more focused on few individuals than profile page views. Backstrom et al. observed a gender homophily: "We find that females send 68% of their messages to females, while males send only 53% of their messages to females. This distinction is consistent with gender homophily — in which each gender has a bias toward within-gender communication — modulated by the overall distribution of Facebook messages. On the other hand, we see much smaller differences in viewing: for typical activity levels, both females and males direct roughly 60% of their profile viewing activity to female users." In contrast to Backstrom et al. [Backstrom et al., 2011], we evaluated user behavior patterns more detailed with respect to timings and content contribution and consumption. We further evaluated historical and device usage independent data to understand the development of user behavior over time in the recent past.

2.8 Summary and Conclusion

In this chapter, we presented a study on user behavior in Facebook with 2,071 participants. We elaborated the function usage to find out which is the dominating function in Facebook, we evaluated the newsfeed and depicted what kind and how much information is shared amongst which actors. Since Facebook is widely known as a tool to communicate with friends, we checked this assumption by explicitly elaborating how often the study participants communicated with their friends using a certain functionality. Finally, we elaborated the dynamics in user behavior, since research on user behavior means to learn about a moving target.

Facebook's major strength is its efficiency in simplifying communication. First, it leverages cross channel effects. Facebook users only need to add persons as friends to stay in touch. Lists of friends act as IDs for many communication channels such as messaging, content sharing or even voice chats. Second, it automatically filters stories in the newsfeed to allow users to efficiently grasp the most important information about their social environment. From content producer's point of view, this filtering leads to an automatic audience selection. Third, its friend recommender system simplifies finding social contacts. Fourth, it is time-efficient to send birthday congratulations. Facebook thus becomes the channel through which people receive attention from their social contacts.

The evaluation of our dataset yielded several different findings. We summarize them as follows:

- Sessions in Facebook are shorter and less frequent than assumed in the literature. In particular, very long sessions are missing.
- The newsfeed is the most intensively used function in Facebook.
- Content contribution is very disparate. A few users contribute a major share of content.
- FPA users consume many items in a short time per day (average: 43 items in 6:44 minutes).
- Shared content in Facebook is very fresh. 84.79% of all posts are not older than 24 hours until being shown to the recipients.
- The probability of a commercial newsfeed entry to be viewed roughly equals those of friend's posts in average.
- User behavior in Facebook is changing at the scale of years. While low effort actions, such as likes and reshares, recently became more popular, the contribution of photos, status updates and comments is relatively decreasing.
- Facebook became mature and stable. This is reflected not only by decreasing user growth rate but also by decreasing establishment of new connections amongst the existing set of users. Users discover fewer new people to add as friend.

We conclude from our observations that users recently seem to use Facebook with a higher speed and lower effort than before, preferring quick actions with low commitment (e.g. likes and reshares). Also, users prefer extremely fresh content. As a consequence, alternative OSN architectures, such as P2P-

based OSNs (e.g. [Cutillo et al., 2009c, Narendula et al., 2012, Shahriar et al., 2013, Buchegger et al., 2009a, Paul et al., 2014a]), could be designed in a lightweight way without the burden of persistently storing stale content in large user profiles. Because of their low storage overhead, the large fraction of shared (external) links in the newsfeeds supports the idea of small user profiles, too. Furthermore, focus of alternative OSN architectures should be brought on dynamic environments, caused by short session durations.

This work has also highlighted the dynamics in user behavior at a scale of years. We assume both technological influences, such as advances in the sector of mobile computing, as well as the social reasons, e.g. learning curves of users and privacy discussions, to be drivers of dynamics in user behavior. User behavior in OSNs thus should be studied while being aware of these dynamics. Stale user behavior models should be carefully used to evaluate novel systems, since stale models may not reflect the recent situation in OSNs.

Surprisingly for us, users spend a major share of their attention and time with commercial pages. The probability of commercial newsfeed entries to be viewed roughly equals those of friend's posts (does not hold for a small set of close friends). Thus, Facebook seems to be successful to target the recipients of commercial news and it seems to insert a compatible amount of commercial content into the newsfeed.

Chapter 3

Privacy Preferences of Facebook Users

OSNs allow their users to create and maintain a personal user profile and connect this profile with others by declaring friendship relations. Amongst communication functionalities, sharing content and personal information is the core of OSN sites. Content sharing serves communication and self-expression needs of OSN users, but raises privacy concerns at the same time.

There is an ongoing discussion about how to handle those privacy concerns. The CEOs of Google and Facebook argue that we live in a post-privacy world[1,2]. We shall accept the fact that there is no privacy anymore and adapt ourselves to the new situation. On the other side of the discussion spectrum, privacy advocates fear oversharing of content [Liu et al., 2011b] to avoid adverse effects such as that employers are accessing private information to draw undesired conclusions. In spite of this discussion, the real privacy preferences of the social networking community are still not entirely known.

Studying the actual privacy settings of Facebook users does also not tell the whole story about content sharing and privacy preferences (e.g. [Krishnamurthy and Wills, 2008]), since users are commonly unable to select the desired audience [Liu et al., 2011b, Madejski et al., 2011]. We thus developed a color-based interface to simplify the audience selection for user content in Facebook. It reduces both errors and effort of choosing the audience. Subsequently, we developed and published an extension for the Firefox and Chrome browsers to help users to meet their privacy preferences and to study the latter.

In this chapter, we study the privacy preferences of users in Facebook and present our first approach to mitigate adverse effects in OSNs. To that end, we present our audience selection interface, together with two user studies. The first user study shows the effectiveness and efficiency of our audience selection interface in a controlled en-

[1] http://www.theguardian.com/technology/2010/jan/11/facebook-privacy, accessed on 2015-11-01

[2] https://www.eff.org/deeplinks/2009/12/google-ceo-eric-schmidt-dismisses-privacy accessed on 2015-11-01

vironment. The second study in this chapter is a large-scale user study with 4,182 users from 102 countries that shows the impact of the new interface. We evaluate the behavior from real users who perform audience selection on their own user profiles for their own reasons on their own devices.

3.1 Reducing Maloperation Risks in Audience Selection

Previous research concordantly argues that privacy enhancing technologies, including distributed and secure data storage are important for OSN [Shakimov et al., 2009, Tootoonchian et al., 2009]. Yet, it can only improve the situation if the users are actually able to choose the adequate audience for their postings and properly configure their privacy settings accordingly. Furthermore, there is consent that this can only be ensured by increasing intelligibility of current privacy controls [Madejski et al., 2011, Liu et al., 2011b]. To this end we propose *C4PS - Colors for Privacy Settings*, a novel concept for privacy settings and their representation. *C4PS* aims at minimizing the cognitive overhead of the audience selection process, based on three foundations:

- color coding of authorization settings with immediate feedback upon change,
- one-click configuration based on proximity of data and respective controls and
- group-based access control through aggregated configuration, and easy group management based on drag-and-drop.

While we implemented and tested *C4PS* as a proof of concept for Facebook, the idea is generally applicable to any OSN, or other web pages with privacy settings.

We started with a *C4PS* mockup for the Facebook interface early 2011 to evaluate, if *C4PS* indeed simplifies the authorization task and performed a lab user study. The results

- indicate that modifying and inspecting the privacy settings is significantly easier and more efficient when applying *C4PS* and
- confirm previous studies showing that even users who consider themselves proficient with the Facebook site are unable to correctly perform precise privacy settings.

Based on the results of the study we provide a Firefox plug-in applying *C4PS* to the modified Facebook interface after the introduction of the *Timeline* for download.

The rest of this section is organized as follows: We present the rationale concept and design of *C4PS* in Section 3.1.1 and the color scheme and interface usage in Section 3.1.2. The methodology of our user study is described in Section 3.1.3 and its results in Section 3.1.3. We conclude the study with a summary and future work in Section 3.1.4.

3.1.1 *C4PS:* Design Principles

The concept of *C4PS* is based on four main principles. The first three cover usability aspects according to ISO 9241, and the last one defines the applicability of the interface.

P1 - Little Effort: To ensure high accuracy when working with the interface, the user shall be able to check or change his privacy setting with as little effort (easy and fast) as possible (inspired by ISO 9241-11 – effectiveness and efficiency; and [Krug, 2005]).

P2 - Applying Common Practices: To minimize the learning effort while becoming accustomed to our interface, commonly accepted and well-known usability patterns shall be used to support users – like colors, drag and drop, tooltips or graying out inactive elements (inspired by ISO 9241-10 – conformity with user expectations).

P3 - Direct Success Control: To avoid gaps between intended and actually performed adjustments (as shown in [Madejski et al., 2011]), results of modifications to the privacy settings shall be displayed and visible instantly (inspired by ISO 9241-10 – self descriptiveness).

P4 - Applicability: To cause the least possible cognitive overhead for accustomed users and to stay independent of Facebook, *C4PS* needs to allow for direct integration into the existing web pages.

Based on these four principles, we developed concepts for *C4PS*, identifying a need for new functionality for both the main privacy settings as well as the group management.

3.1.2 *C4PS:* Color Scheme and Interface Usage

Regarding the main privacy setting functionality we highlight each attribute in the profile by a particular color, depending on the group of people who are granted access. We also enable the user to change the accessibility with just one click, support the group selection with tooltips, make this privacy settings mode easily accessible, and provide very brief instructions. In addition, the privacy settings mode provides a button to check how others see the profile. These concepts are explained in detail in this subsection.

Color Coding: The colors used are guided by the well-known traffic light colors (*P2*). Blue was added to represent custom settings. The corresponding color definitions are:

- *Red:* Visible to nobody
- *Blue:* Visible to selected friends
- *Yellow:* Visible to all friends
- *Green:* Visible to everyone

All privacy settings are visualized by our color scheme in the *C4PS* privacy setting mode (*P4*), so that an attribute's visibility can be directly derived from its coloring (cmp. Fig. 3.1). The concrete choice of colors (e.g. green to represent public visibility) was subject of both: internal discussions and our user study. We will later present the preferences of study participants.

Easy To Modify Setting for Single Attributes: The user can change the privacy setting for a specific attribute by simply clicking the buttons on the edge of the row on the right side (*P1*). The color of the buttons shows the visibility that will be set for the entry by clicking on it (e.g. in Fig. 3.1). The settings are changed immediately (*P3*), which is reflected directly by a color change of the attribute's cell. If the user chooses "selected

friends" (blue), a window opens in which friends or groups are granted access to the mentioned attribute.

Tooltip: To further increase the usability, tooltips indicate the setting corresponding to the color for each button (*P2*). Tooltips are shown when the mouse courser hovers over the button (cmp. Fig. 3.1).

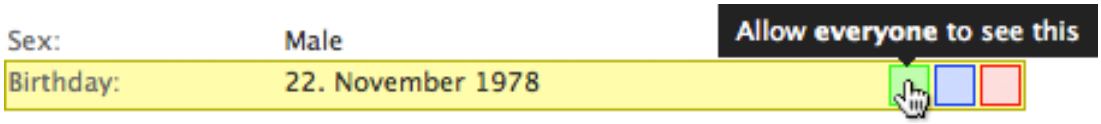


Figure 3.1: Color coding for one attribute - birthday

Easy Access to Privacy Settings: C4PS integrates in the mockup a new button under the profile picture to enter the *C4PS* privacy settings page. This button is visible on each Facebook page and thus the *C4PS* privacy settings page is easy to access (*P1*). After switching to the privacy editing mode and editing the privacy settings, the user can exit this mode by clicking a button labeled "Stop editing privacy settings" at the same place. In the improved version we enabled the visibility of color coding instantly without entering any privacy settings mode.

Information on the Top of the Page: According to common practice (*P2*), general information about the color visualization and the meaning of each color are provided on the top of the page in the editing mode.

Check how the own Profile is Shown to other Users: The privacy settings mode provides a button at the top of the page 'How others see your profile', which offers a simple visualization to check how selected other people - including friends - see the profile (*P1*).

Application to Photo Albums: The privacy settings for photo albums can be checked and modified with the same color mechanism. When visiting the Facebook "photos" tab, an overview of all photo albums of the user is displayed, as in the original Facebook interface. However, there is an additional button labeled "Edit Privacy Settings" (cmp. Fig. 3.2).

This button again activates the *C4PS* privacy editing mode. Here, the photo album elements are highlighted with a color indicating the privacy setting (cmp. Fig. 3.3). Additionally, three colored buttons are shown on every item and allow to change the privacy setting as described before. Clicking on the colored buttons changes the privacy setting for the entire album, while individual restrictions, set to single photos, remain unchanged. To change the privacy settings of a single photo the user can open the photo album, in which the colored privacy buttons are placed at each photo.

With *C4PS*, checking and modifying privacy settings in Facebook takes a minimum of two steps:

1. Accessing the *C4PS* privacy settings main page by clicking on "Edit Privacy Settings".

2a To inspect the current settings for the profile entry, the user only needs to properly interpret the color. In case of custom settings a third step is required.

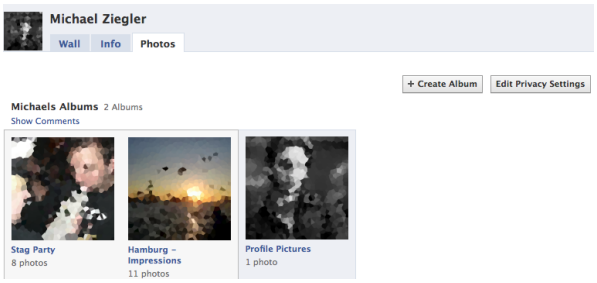


Figure 3.2: Photo albums without privacy settings

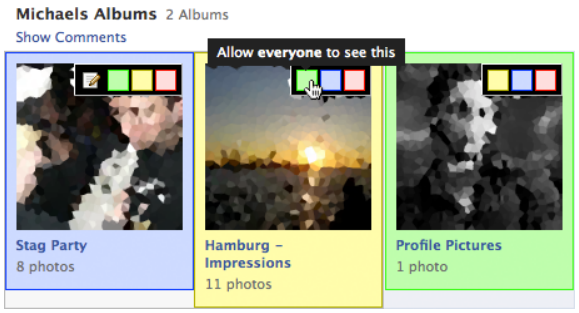


Figure 3.3: *C4PS* interface - photo albums

2b To change the setting of any attribute, the user can simply click on the button colored accordingly.

3.1.3 User Study

To evaluate *C4PS*, we conducted an extensive, controlled lab study. We aimed at validating the following four hypotheses:

H1 *C4PS* makes it easier and faster to find out to whom a particular attribute is visible.

H2 Using *C4PS*, testing how the complete profile is presented to another user is easier and faster.

H3 Setting the visibility of attributes is easier and faster using *C4PS*.

H4 The group management can be handled easier and faster using *C4PS*.

These four hypotheses are intended to cover all aspects that may concern users aiming to adjust their privacy settings. In addition, we were interested in the feedback about the concrete ideas implemented in *C4PS* to further improve it.

We decided to run a lab study because this enabled us to measure time and clicks while the participants solved some tasks with both interfaces - the improved one and the original one. Correspondingly, the participants were asked to use a lab PC and a Facebook profile that we created, to set a controlled environment without requiring the participants to disclose their own Facebook profiles.

Course of Action

The study contained the following phases:

1. Answering the OSN questionnaire (provided on paper), containing eleven questions (see appendix 8.3) regarding the use of OSN in order to estimate the prior knowledge of the test person
2. First practical part, during which several tasks have to be solved with one of the interfaces. Note, to prevent a possible learning effect due to the first use of one of the two interfaces, the order of presentation of the two interfaces was alternated for each participant. The answers had to be written down (on paper).

3. Answering the "System Usability Scale" (SUS) questionnaire (provided on paper) as introduced by Brooke [Brooke, 1996]. It allows measurements concerning effectiveness, efficiency and user satisfaction, and due to its generality is applicable to various types of systems.
4. Second practical part for solving the tasks with the second interface.
5. SUS questionnaire was applied to the second interface
6. Answering the usability questionnaire (provided on paper), containing 15 questions regarding the usability of the new interface and a field for general comments (comp. appendix).
7. Answering demographic questions (provided on paper) concerning age, gender, and profession.

Nr.	In the practical part of the study, we asked the test persons to:
1.	Find out to which users or groups the birthday (Task 1) / hometown (Task 2) / relationship status (Task 3) / a particular photo album was visible (Task 4)
2.	Find out which attributes were visible for a specific friend (Task 5)
3.	Create a group "best friends" (Task 6)
4.	Add two particular friends and the group "class mates" to the group "best friends" (Task 7)
5.	Adjust the privacy settings of five attributes - mobile phone number to only two specific friends (Task 8.1) / interests to all (Task 8.2) / hometown to only one specific group (Task 8.3) / relationship to no one (Task 8.4) / religious and political views to all friends (Task 8.5)
6.	Adjust the privacy settings of one selected photo album, granting access to a specific group, except a single particular friend, being part of the group (Task 9).

Table 3.1: Tasks for participants to solve during our *C4PS* study

All tasks (Table 3.1) had to be solved in this particular order while it was not required to start from the main page after login. This course of action is more realistic, as users usually want to check or edit the privacy setting for more than a single attribute.

Evaluation Criteria

The following information was deduced from the screencast:

- *Time:* The time a test person needs to perform a task. This measure is used to compare the efficiency for users in solving tasks.
- *Clicks:* Number of clicks a user needs to complete a task
- *Success:* The task-solving success of a study participant. It is only distinguished between the values 1 (task solved completely and correctly) and 0 (failure to precisely solve the task). The success rate per task measures the fraction of users solving a task with success.

The measurement of time and clicks for a task was performed manually. The first goal-directed mouse movement was the starting point for the measurement of a task. The end of the measurement was chosen to be the successful or failed completion of a

task, or the user canceling the task. We used the time frame without mouse movement before a new task was started as an indicator for canceling. We did not count clicks incidentally placed beyond any button or link as well as multiple clicks on a button or link to start a function (while waiting for the website to respond). This should preserve the comparability of values. All other clicks to perform a task were counted. This includes clicks on scroll bars, selecting text or clicking into input forms. The time and clicks between tasks was stripped.

To evaluate our hypotheses, we measure both the time and clicks it takes to solve a task to evaluate if a system is *easier and faster*, and we consider the success of a task solution. The usability questions from the SUS questionnaire, Attrakdiff® questionnaire, and our final own usability questionnaire additionally are taken into account to gauge intelligibility and acceptance of *C4PS*.

Sample Description

In this section, we briefly describe our study sample with respect to demographics as well as participant's experience with OSNs. This allows to judge the limitations of our study.

The study was performed with 40 students, aged between 20 to 32 years. Recruiting was done in lectures and via email lists. The information provided to the participants was that a new interface for the privacy settings in Facebook would be tested. Participants were rewarded with sweets.

All participants were members of at least one OSN, except for three participants. 57.5% access their OSN profile(s) at least once a day and 25% even several times a day. Nearly two thirds of the study participants are Facebook users. Almost all probands (90%) have already been in touch with the privacy settings of their OSN provider. However, many of them consider these settings to be confusing (57.5%). 15% of the participants were very concerned about their privacy settings and stated that they modify or check them every month. The rest did it less often. 30% did not change the privacy settings, after they have been set up once. The possibility to create lists or groups of friends was only used by 27.5% of the participants and the possibility to set certain rights for groups or for individual friends was used by 37.5%. 62.5% of the participants stated that they are aware of the visibility of their profile's attributes to other network members.

Expectations

While designing the user interface and the user study, we expected the new interface to help participants to solve the tasks better (less mistakes) and with lower effort (less clicks). In particular, we expected less OSN savvy probands to gain the largest benefit from the new interface.

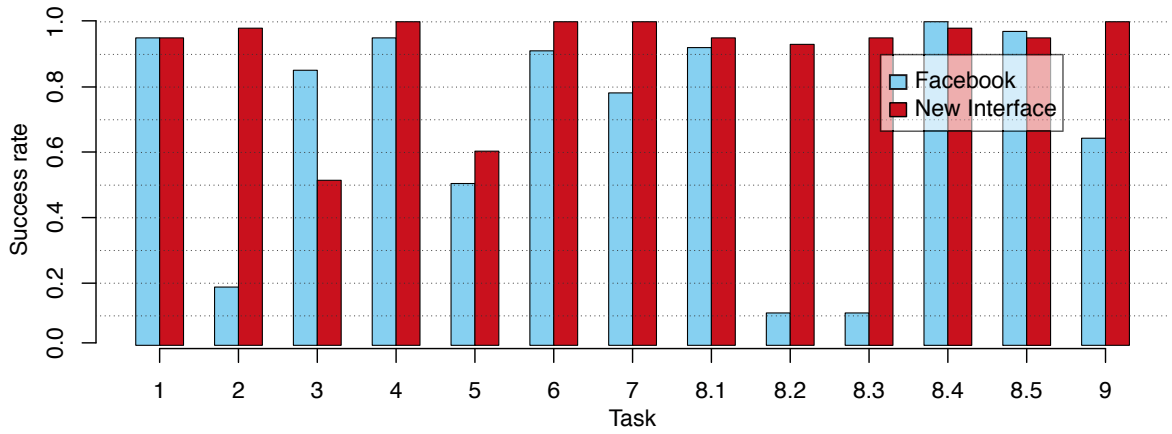


Figure 3.4: Success rate per task

Results

We first provide the results of the study regarding success rate and efficiency. Afterwards, we discuss the feedback regarding the three usability questionnaires (Subsections 3.1.3 and 3.1.3). We show that the four hypotheses can all be confirmed in each category according to the evaluation criteria defined in Subsection 3.1.3. Based on these results we provide some ideas for further improvements.

Success Rates and Efficiency Analysis

In this subsection, we show that the four hypotheses hold regarding the success rates, the time needed, and the number of clicks needed to complete the corresponding tasks.

Success Rates: The overall success rate for all tasks and all participants in the new interface is 91% while it is only 68% for the original Facebook interface. As shown in Figure 3.4, the success rate for the new interface is higher than the one for the original interface in almost all tasks. Only for task 3, the original Facebook interface leads to better results. Here, subjects were asked to list the friends or groups who have access to the attribute "Relationship Status". Unfortunately the participants wrote down the privacy setting "selected friends" while we expected them to read out the actual list of friends who have access. In most cases, the participant did not click on the blue button in order to get this information but only wrote down the tooltip text (selected friends) that was revealed when hovering over the button. Some other participants did not write down all groups having access to this attribute or the wrong ones. According to our definition, both cases were interpreted as wrong answers.

The biggest difference was measured at task 2 (visibility of the field "current city and hometown"). Only 17.5% of the participants solved this task correctly with the original interface, while all but one participant succeeded using the new interface. One reason for this is that - using the original Facebook interface - this attribute is placed in the "Connecting on Facebook" section rather than on the main privacy settings page. In addition, many participants wrote down the value of the incorrect attribute "Contact information", which was displayed on the main privacy settings page on Facebook.

The difference between both interfaces again is very large for Tasks 8.2 and 8.3, for a similar reason, and the participants hence changed the wrong attribute. For Task 8.3, participants changed the field "Contact information" instead of "hometown" while for task 8.3 the incorrect attribute "Interested in" was changed, instead of "Interests". The latter in this case represents the gender the user is interested in rather than the intended interest in his activities like sport, films, music or other.

Efficiency Analysis. The efficiency analysis with respect to time and clicks below compares only Tasks 5 to 9, since in these tasks the participants actually had to change settings, rather than interpreting the current configuration.

The minimum number of clicks to properly execute Tasks 1 to 4 using *C4PS* is one click on "Edit Privacy Settings" from the main page, then interpreting the privacy settings for the first two requested attributes. In the case of Task 3, a further click was required, as the displayed privacy level "selected friends" was not the proper answer, but it was necessary to interpret which selected friends were granted access by clicking on the blue button. Thus, one click was necessary to open the dialog, and another one to close it. Similarly, it was required to click on the photo album settings to discover this information. The minimum number of clicks in *C4PS* thus amounted to four. The minimum number of clicks to execute these tasks properly in Facebook amounted to eight.

Time needed: Most tasks were completed faster when using *C4PS*, as shown in Fig. 3.5. Especially when adjusting privacy settings that are in the "Connecting on Facebook"-category and while creating groups. The test users on average need more than twice as much time to solve the tasks using the Facebook interface, as compared to *C4PS*. Fig. 3.5 also shows that the variance using *C4PS* is much lower for most tasks, indicating that all users achieved approximately the same efficiency.

Clicks needed: Considering the number of clicks (Fig. 3.6), the results are very similar to those from the time measurement. Most tasks can be solved with much fewer clicks using *C4PS*, and the variance is very low. The participants generally needed nearly three times more clicks to complete the task using the original interface. Note, that it can be assumed that a much greater deviation would have been achieved, if all privacy setting tasks had to be performed separately starting from the main menu. Using the Facebook interface, the user would have needed to perform at least three additional clicks to get to the settings menu, compared to a single click that is necessary using *C4PS*.

Comparing users with and without Facebook accounts. The probands who already use Facebook had an advantage when solving the tasks, because they already knew the look and feel of the Facebook interface, or even the concerning privacy settings. However, even those participants achieved better success rates with *C4PS*, even if they could be considered Facebook experts for using it every day. In numbers, the success rate of Facebook experts for the tasks on Facebook was 73% compared to a success rate of 94% when using *C4PS*. Participants who were not considered Facebook experts only reached a success rate of 60% for the task when using the original interface, rising to a success rate of 86% when using *C4PS*.

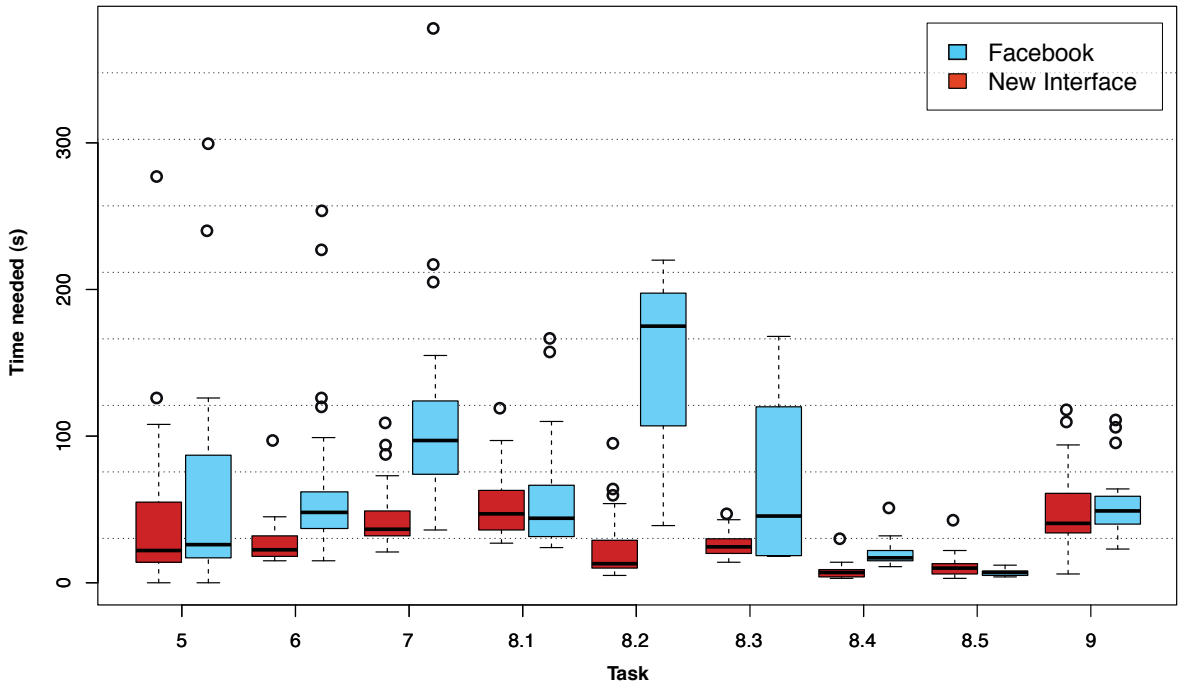


Figure 3.5: Required time per task

Almost all tasks have been solved better by participants that are Facebook users (in both interfaces). Solving the task on Facebook, the experts needed 1.65 less clicks on average. When using *C4PS*, the disparity between experts and normal users was smaller. The experts in this case completed the task with 0.89 less clicks. Measuring the time for completing tasks, the experts performed 1.76 times faster using Facebook. Using *C4PS*, however, the experts were only 0.75 times faster. This disparity shows an additional improvement of the usability of the systems, and the subjects who had not used Facebook before had a much harder time to cope with the original interface at all.

The results for all three criteria show that even users who consider themselves proficient with Facebook are unable to correctly perform precise and efficient privacy settings.

SUS - System Usability Scale

The System Usability Scale (SUS) [Brooke, 1996] is a popular one-dimensional psychometric scale (range: 0-100) that allows to measure and compare the usability of systems. It is determined by a standard questionnaire with ten questions and can be applied in a vast variety of contexts.

Referring to A. Bangor et al. who analyzed nearly 1000 SUS studies [Bangor et al., 2009], acceptable products have a SUS-score of over 70. Better products start at the high 70s and end in the upper 80s range. Only truly excellent products have a score above 90. Products with scores less than 50 should be cause for significant concern and are judged to be unacceptable. Due to this scale, the usage of our interface is very good while Facebook itself reaches numbers below those for acceptable products.

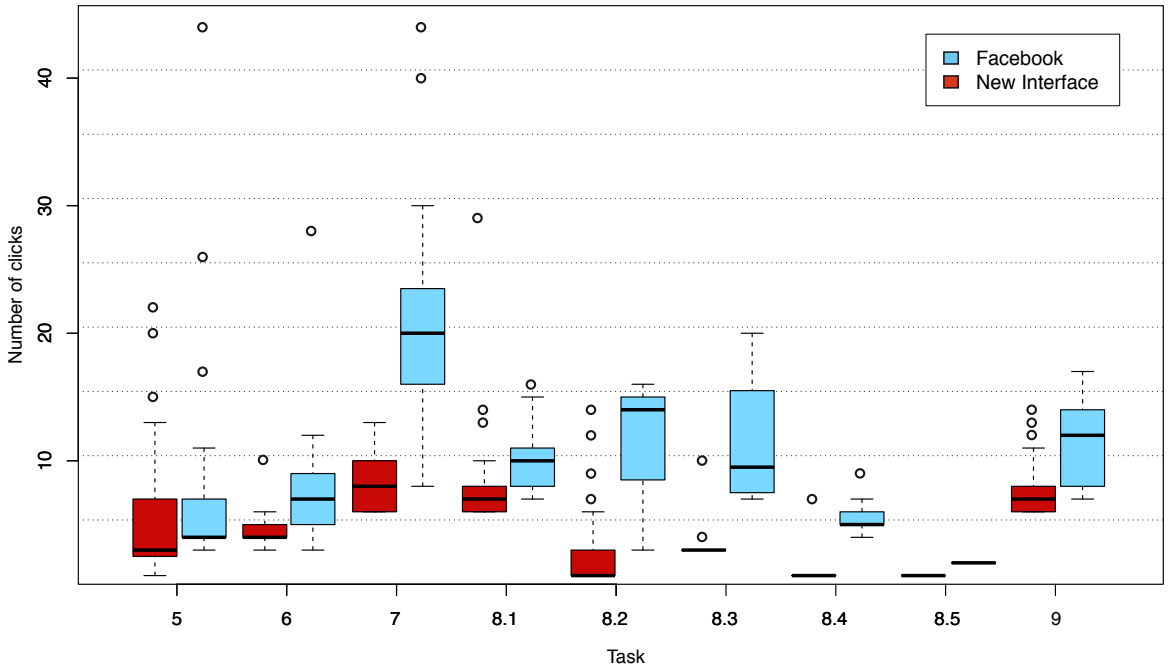


Figure 3.6: Required number of clicks per task

The average SUS value for our interface (all users) has been evaluated to 82.6. The maximum possible SUS value of 100 was achieved at maximum, and the worst rating of the interface was valued at 37.5. Comparing this with Facebook, the users rated the interface with an average SUS value of 35. The maximum value was 75 and the minimum was five. That means that our participants rated *C4PS* to be very good in terms of usability while the original Facebook audience selection was rated to be poor.

Participant's Opinions about *C4PS*

At the end of the study, we asked the study participants what they like and do not like as well as what they would improve. The results of this questionnaire are discussed in this subsection. They show that *C4PS* also performs better regarding these interface specific usability questions, and that people like the general concepts.

57.5% of the participants rated the original Facebook privacy setting mechanisms as confusing (the worst level on a scale of four possibilities) and only one stated that it is very clearly arranged (the best level). 87.5% of the participants stated that *C4PS* improved the situation a lot (maximum improvement of a scale of four options). On a scale with 4 options 50% rated the visualization with colors as very good, 47.5% with good and the rest with level 3 while no one selected level four. The question whether the color coding is well-defined was agreed by 31 (77.5%) of the participants.

Only 20% of the participants answered that they cope 'very well' or 'well' with the original interfaces for group management while 97.5% of the participants made this statement for the new interface. The question regarding the usability of the privacy setting mechanisms was answered with 'very good' by 5% of the participants for the

original Facebook interfaces and by 47.5% for the new interfaces while 22.5% (Facebook) and 50% (new interface) stated that these mechanisms in the corresponding interfaces provide a 'good' usability.

There were also two fields to provide comments. In the first one we asked the participants what they liked most about the interface. Almost everyone mentioned the colors while only a few also mentioned the group management. People stated for instance that the privacy settings are 'easy', 'clearly arranged', 'directly accessible', 'easy to find', 'easy to use', 'everything is on one page', 'less clicks', 'quick', 'applicable for more attributes' and 'clear what to do'. In the second field we asked them to propose further improvements. Comments mainly addressed the group management and the profile preview in general and for the case that particular friends have the right to access this attribute. Some remarks were made regarding the colors - including only three colors, changing colors, self-defined colors; and also the fact that the order of the colored buttons in a row should stay the same.

In spite of the subjective nature of the participant's opinions, presented in this section, these details are meaningful. They underline that - in contrast to Facebook's original audience selection interface - the study probands had a positive experience with *C4PS*. We assume that these positive experiences to encourage users to be engaged in their privacy settings.

3.1.4 *C4PS:* Mock-up Study Summary

Even though users publish highly personal data on OSN sites, several studies have shown that they are incapable of configuring their privacy settings correctly. The direct consequence is unwanted over-sharing of highly personal information by the users, which allows for various attacks, including information harvesting and various types of social engineering.

To increase the intelligibility of the authorization controls, we have proposed, evaluated, and implemented *C4PS – Colors for Privacy Settings*. *C4PS* introduces a new mental model for the privacy settings, and has been designed as simple and intuitive as possible, to minimize the cognitive overhead of the authorization task. It is based on the foundations of color coding, simple, one-click configuration, and group-based access control, including a simplified group management interface. We initially implemented *C4PS* as a mockup for controlled lab studies.

Evaluating *C4PS* in an extensive, controlled user study demonstrated two main insights:

1. *C4PS* greatly aids the authorization steps – it not only enables the user to grant exactly the desired authorization, but additionally helps the user comprehend their authorization activities and current settings.
2. Even users who are convinced of their expertise using Facebook are unable to employ the existing privacy controls correctly and efficiently, and are unable to precisely configure their profile according to the desired authorization.

Based on the observed success rates, we argue that *C4PS* enables OSN users to properly choose the desired audience. We implemented *C4PS* in Firefox and Chrome

browser-extensions (plugins), which are available for download from our web site. This browser extension is called Facebook Privacy Watcher (FPW). The latter enables us to understand privacy preferences and over-sharing in OSNs by deploying large-scale user studies.

3.2 Large-Scale User Study on Content Sharing Preferences

The FPW received huge attention from international media, such as newspapers, radio stations and blogs. As a consequence, it was downloaded more than 44,800 times from at least 102 countries. We asked the FPW users to join this user study by sending us anonymized feedback with consent to improve the plug-in and to evaluate the impact of the plug-in on user's privacy. We received 9,296 feedback responses originated from 102 countries. These responses included the privacy settings of the user profiles and the changes that were made with the help of our plug-in. Furthermore, we received the number of friends, photos, likes, notes and map entries as well as the binary information for each user profile data field (denoted profile field in the remainder) whether it is filled with data or not.

Based on this dataset, we evaluate the real exposure of private user data in Facebook and the content sharing desiderata of the FPW users. We evaluate the privacy settings before and after introducing a comprehensible visualization of privacy handle as well as the changes that have been performed. Since both, the privacy settings as well as the user profiles profiles (e.g. the number of photos), strongly differ with respect to different countries, we also performed evaluations that focus on national differences. Assuming that increasing or decreasing the visibility of parts of the user profiles expresses the desires of users to have more or less privacy, we compared the user profiles of users who use the FPW to achieve more privacy with those who decided to publish more private data.

Our results show that users intentionally hide content from being publicly accessed and do not accept the default privacy settings even before using our plug-in. With the help of the FPW, users hide critical data fields such as friend lists and family member markers but publish birthdays and religious views. The total amount of content which is visible to Facebook users does not dramatically decrease after introducing a comprehensible visualization of privacy controls, but the composition of the visible content changes. The content sharing patterns are strongly depending on their country of origin.

The contributions of this user study are (i) to provide an understanding of the content sharing preferences of FPW users both in general and (ii) with respect to different countries and (iii) to explain and quantify the effect of improved usability of privacy interfaces on privacy settings. Since we assume user's privacy preferences to affect both the content sharing affinity and the privacy settings, we further (iv) depict relations between these two aspects of Facebook usage by means of cluster analyses. An important highlight of this study is that we are not limited to public-available data. Due to the FPW feedback data, we can take the user profile owner's point of view on her privacy settings.

The remainder of this section is organized as follows: We first provide a detailed data description in Section 3.2.1. In Section 3.2.2, we evaluate the privacy settings of FPW users and the impact of introducing a comprehensible audience selection without mentioning country specific differences. Because of vast differences amongst users from different countries, we provide a deeper analysis of those specifics in Section 3.2.3. The relation between sharing desiderata and quantifiable user profile properties such as the numbers of friends, likes and photos are evaluated in Section 3.2.4. We discuss the related work in Section 3.2.5 and summarize our findings and conclude our work in Section 3.2.6.

3.2.1 Experimental Setup and Dataset Description

In this section, we specify the setup of our study by describing our ethical considerations and the precise data collection methods. To underline the adequacy of our color-coding audience selection interface to be used in this study, we describe essences on the feedback from study participants. We further depict which and how much data we were able to collect in this study and describe basic user profile statistics of the participants. The bias as a result of a non-random selection of study participants is also discussed in this section.

Ethical Considerations

We protect the privacy of our study participants! Neither the download logfile which we used to estimate the dissemination of the FPW, nor the feedback answers that we collected are linked to individuals. We asked the FPW users to send us feedback with consent. We explained the reason for collecting the data and allowed users to access and verify the data before sending it to our server. All feedback responses that we used in this study are anonymized. We keep the collected data confidential to protect all study participants from deanonymization attempts and do only publish aggregated data.

Users' Acceptance of the FPW

It is essential for the success of the study that participants are willing to integrate the tool in their normal OSN usage and to use it more than once. The FPW and the realized user interface hence need to be both: beneficial for the participants and easy to use. Thus, the first question which we asked our users in the feedback formula was: "How do you like the idea to use colors to visualize privacy settings?".

The overwhelming majority rated this idea as "very good" (65.66%) or "good" (32.2%). Less than one percent rated the idea to be "medium" (0.98%), "bad" (0.46%) or "very bad" (0.7%). However, the plug-in did not work from 7th of November 2013, 2:30 am, till 8th of November, 3:30 (am, CET), due to Facebook site changes. During this time, we received most of the negative ratings.

Creating the color scheme, we argued in the team which type of color scheme is more intuitive to the users: green, inspired by traffic lights meaning "go" - corresponding in

Rating	Percentage
Very good	32.34
Good	61.34
Medium	3.44
Bad	1.83
Very bad	1.05

Table 3.2: How do you like the FPW implementation?

the color scheme to be visible to everybody or green in the meaning of being safe since the item is not visible to anybody. This question has been asked in the previous user study with 40 participants. 60% of the participants preferred the green to represent the setting meaning 'visible to everybody'. It roughly meets the results in this study (54.83% vs. 45.17%).

The implementation was not rated that good like the idea of using colors for setting privacy. Evaluating the comments, we can find the following reasons: First, the people who preferred green to represent the safe setting where nobody has access were not satisfied that it was not possible to customize the colors in the first three versions. Second, we suffered from a bug in the first version that caused many negative ratings.

Data Collection

We gathered data about the FPW from two sources. The first is the download log file at our own server, where the plug-in can be downloaded from. The second source of data is the set of feedback responses which have been sent to us. While the first source gives us insights into the spreading process of the plug-in, the second source allows us to draw a picture of the plug-in usage as well as its impact on privacy settings of the users' profiles.

Download Log

Analyzing the download logfile enabled us to understand the time and locality dimensions of the FPW dissemination. We discovered strong peaks subsequently to the moments of publication in different venues as well as that a large user basis is originated in Germany and Egypt. We further discovered a couple of sites, offering to download our plug-in[3,4,5,6,7]. Thus, we only have an incomplete view on the actual downloads by analyzing our own download log. Some of those alternative download sites publish the number of downloads. Adding the number of downloads from our

[3] http://www.chip.de/downloads/Facebook-Privacy-Watcher-fuer-Firefox_57997141.html, accessed on 2014-12-04

[4] http://www.computerbild.de/download/Facebook-Privacy-Watcher-7834052.html, accessed on 2014-12-04

[5] http://www.netzwelt.de/download/16629-facebook-privacy-watcher.html, accessed on 2014-12-04

[6] http://www.freeware.de/download/facebook-privacy-watcher_64364.html, accessed on 2014-12-04

[7] http://www.soft-ware.net/facebook-privacy-watcher, accessed on 2014-12-04

Country	# Feedback responses
Germany	7,581
Egypt	272
Austria	218
United States	150
Switzerland	147
France	94
Spain	72
Netherlands	62

Table 3.3: The number of feedback responses that we received from the top eight countries

site to those external download counters, we estimate the total number of download to be higher than 44,800, coming at least from 102 countries. One year after our first FPW publication, 11,000 users still followed every update that we offered.

User Feedback

The usual life-cycle of an FPW instance starts with the installation process and resumes with a check of the privacy settings of the own profile during a few sessions (1-5). The plug-in is sparely used afterwards. We asked our users to provide us feedback after activating the plug-in three times, which usually happened within the first days after installation.

We asked for feedback about both the general idea of coloring the profile items to simplify the privacy settings and the implementation of our plug-in. Furthermore, we offered two text fields to enter comments and suggestions concerning the idea as well as the implementation. We explicitly informed our users about the exact (anonymized) data that we collected. From 2012-10-15 till 2014-07-07, we received 9,296 feedback responses from 102 countries that included coloring and log file information.

We collected the following information from our users:

- a hash value of the Facebook - ID (Facebook- UIN)
- the counter (including timestamps), indicating how often the plug-in was activated
- the visibility of each profile field before the first usage of our plug-in happened
- the visibility of each profile field after using our plug-in
- the type and visibility of timeline entries
- the number of friends
- the number of photos and labels
- the number of likes

Furthermore, our server, which gathered the feedback data, ran a script to extract the countries from which we received the feedback.

Sample Bias and Basic User Profile Statistics

We recruited our sample (FPW users) via an announcement on our homepage and by sending press releases to specialized press. We then witnessed a viral spreading process based on word-of-mouth advertising. The attention of mass media such as newspapers[8], radio stations[9] and an Egyptian web portal[10] followed afterwards. In spite of the broad audience of the respective media, the set of participants is by no means random. We decided not to collect detailed demographic informations about FPW users, since this would be inappropriate for a tool that has been advertised to support user's privacy. Instead, we provide technical information such as statistics about the user profiles to allow the sample bias to be appraised:

X	$X = 0$	$\varnothing_X$	$\tilde{X}$	σ_X
Friends	0%	148.75	96	159.53
Photos	3.43%	181.69	32	572.62
Labels on photos	34.45%	20.54	3	64.45
Photo albums	3.49%	10.71	7	20.05
Locations	17.07%	38.68	4	101.65
Likes	10.06%	90.04	36	145.33
Notes	86.94%	1.49	0	19.5

Table 3.4: Basic profile statistics: percentage of profiles without any entry in field X and the average, median and standard deviation of the number of entries in field X

Our median user has 96 friends, liked 36 pages and shared 32 pictures. Many users have just a few friends (Figure 3.7) and a few of them have plenty of friends. The degree distribution of the friendship graph as well as the median number of friends is similar to those of the whole Facebook graph [Ugander et al., 2011]. We interpret this as an evidence that our FPW users are close to the average user with respect to the number of friends.

3.2.2 Global Privacy Evaluation

In this section, we elaborate which data FPA users upload to Facebook and who is allowed to access it without mentioning cultural differences amongst users from various countries to provide a holistic view. We further quantify the impact of the FPW on the privacy settings and compare the standard privacy settings in Facebook with the actual user decisions to quantify the total demand for modifying the Facebook standard privacy setting to meet users' needs.

[8] http://www.handelsblatt.com/technologie/it-tk/it-internet/facebook-privacy-watcher-im-einsatz-gegen-den-daten-kraken-seite-all/7388782-all.html, accessed on 2014-12-04

[9] http://www.ffh.de/news-service/magazin/toController/Topic/toAction/show/toId/3371/toTopic/die-facebook-ampel-fuer-sichere-postings.html, accessed on 2014-12-04

[10] http://www.masrawy.com/news/Technology/General/2012/October/31/5420245.aspx, accessed on 2014-12-04

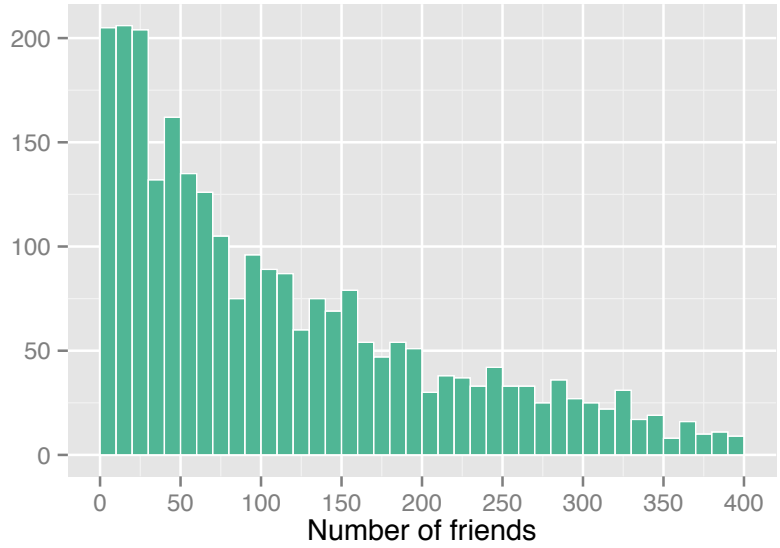


Figure 3.7: Histogram: number of friends

Because of the typical life-cycle of the plug-in instances (Section 3.2.1), three data views are available: the privacy settings before using the plug-in, after using the plug-in and the changes that have been made. We avoid the redundancy which would be caused by presenting the three possible points of view. We instead focus on the settings after applying our plug-in and the changes which have been made.

Exposure of User Data

A Facebook profile can consist of 28 data fields in total. To estimate the potential privacy risk, it is crucial to know which parts of the profile are filled with data and thus potentially exposed to the risk of being accessed by subjects which are not part of the set of desired recipients. The average filling ratio of the profile fields that allow users to select the audience is given in Figure 3.8.

The profile fields friend list, Timeline entries, photo albums, map entries and notes are lists of items that are technically always available. The number of items included in the users profiles can be found in Tables 3.7 and 3.8. Subscriptions are also not included in Figure 3.8. They allow users to follow other users' updates (e.g. news of famous actors) without befriending with them. It is possible to determine the visibility of subscriptions without subscribing anything. According to our ethical considerations, we only store the visibility of data fields but not their content. We thus are not sure whether a user subscribed to any newsfeed.

The fields gender, e-mail and birthday are obligatory to create a user profile on Facebook. Hence, every user profile encloses this data (not necessarily honest). None of the other profile fields are filled by all users. The fields family, current city, relationship status, hometown, employer and school are filled with data by the majority of users. Only few FPW users uploaded skills and phone numbers to Facebook. Please note that we can only check whether data is included or not. We have no means to verify it.

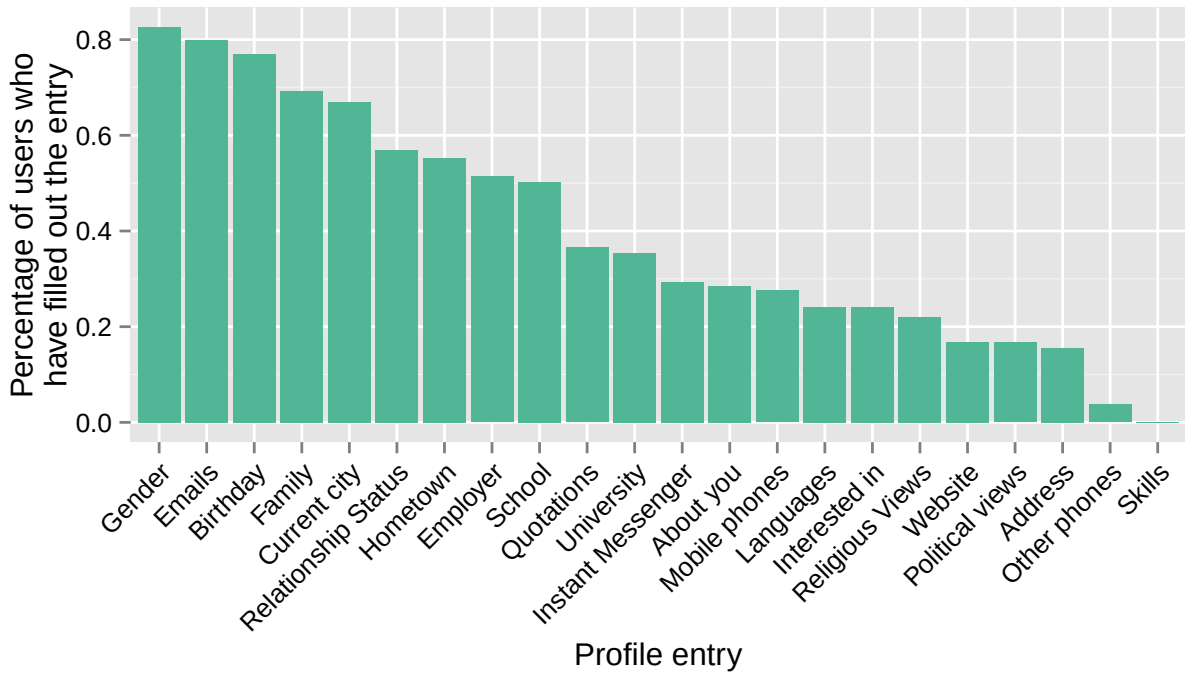


Figure 3.8: Histogram of the ratio, the user profile fields are filled

Visibility of User Profiles Fields

Figure 3.9 shows the cumulated visibility of the profile fields of FPW users. The most popular setting is to share content items with all friends. The second most frequently used setting is to share items with the public. Sharing bits of information with only a subset of friends ('custom') or hiding them ('only me') is not very popular.

More than one third of the users do not restrict access to the fields: current city, employer, friend list, hometown, languages, school and university. These profile fields may help attackers to collect sufficient information to deploy social engineering attacks. The friend list is especially dangerous to publish, since sharing the friend list helps attackers to traverse through the social graph using crawlers. Furthermore, inference attacks [Lindamood et al., 2009] are fostered by publishing the friend list. These kinds of attacks are based on the assumption that friends share similarities (e.g. similar age). An attacker can infer hidden profile attributes in case that friendship connections are known to the attacker and friends disclose the information of interest.

The custom setting is used for phone numbers in more than 95% of those cases where this information is included into the user profile. More than a quarter of our study participants share the birthday, political views and religious views just with a subset of their friends. The fact that a non-negligible number of users use the setting 'only me' is remarkable. It makes sense that people disclose information in fields that are technically necessary (e.g. the friend list) in case that they do not want to share them with others. However, uploading other fields to Facebook without sharing it with anybody does not help to socialize with others. We assume fields with this visibility setting to be a result of increased privacy awareness. Previously visible informations seems to be hidden.

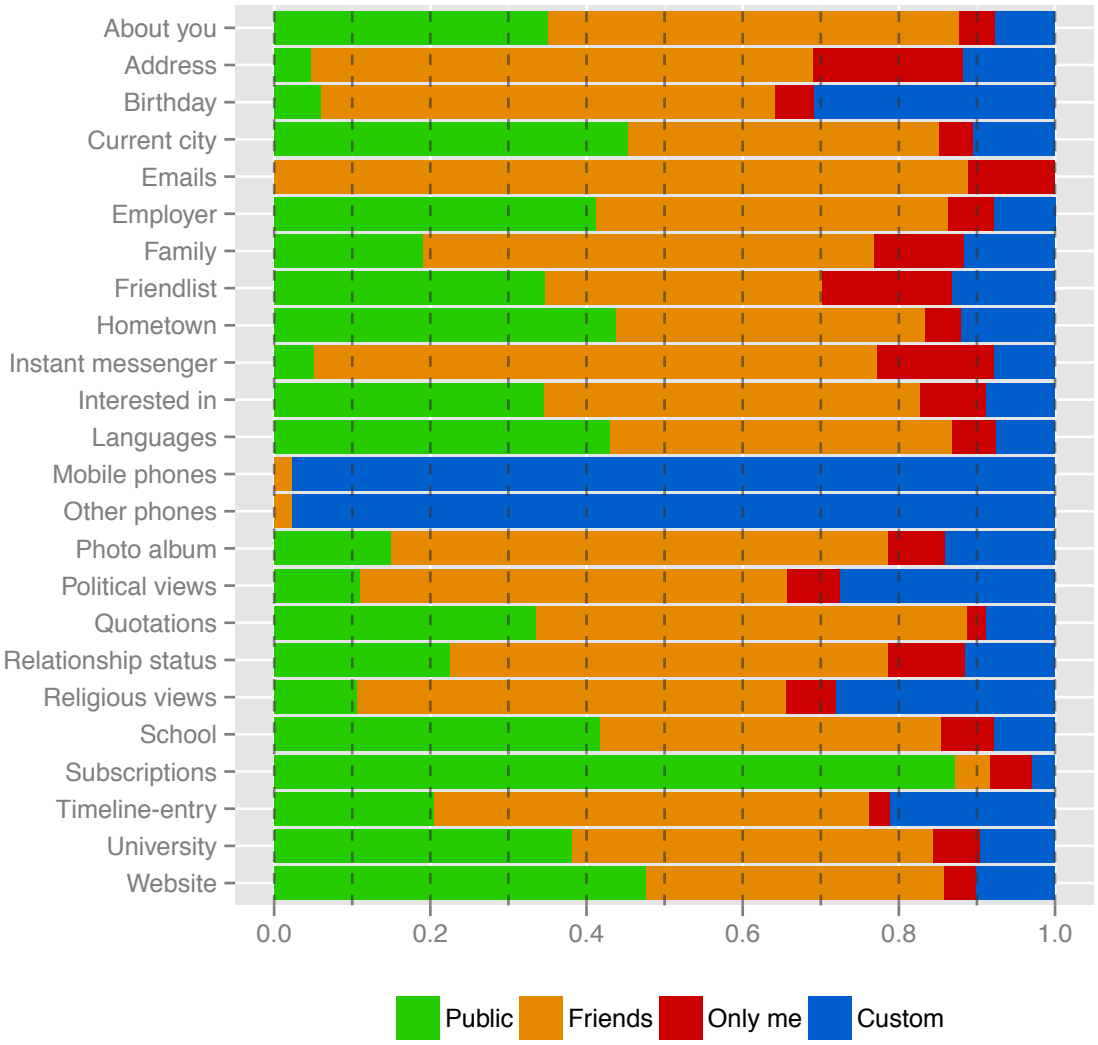


Figure 3.9: Visibility of user profile fields

Timeline entries are similar to posts in a newsfeed and can have many different types. Figure 3.10 shows the visibility of all types of timeline entries. The main findings are that:

- the setting 'friend' is even more dominant than in other parts of the profile
- less entries are visible to the public
- posts from external pages (e.g. commercial pages) and cover photo changes are always public
- the setting 'only me' is rarely used in general
- the most frequently hidden timeline entries are likes from external pages, posts from other users and posts from apps
- photos of other users are often shared with only a subset of friends

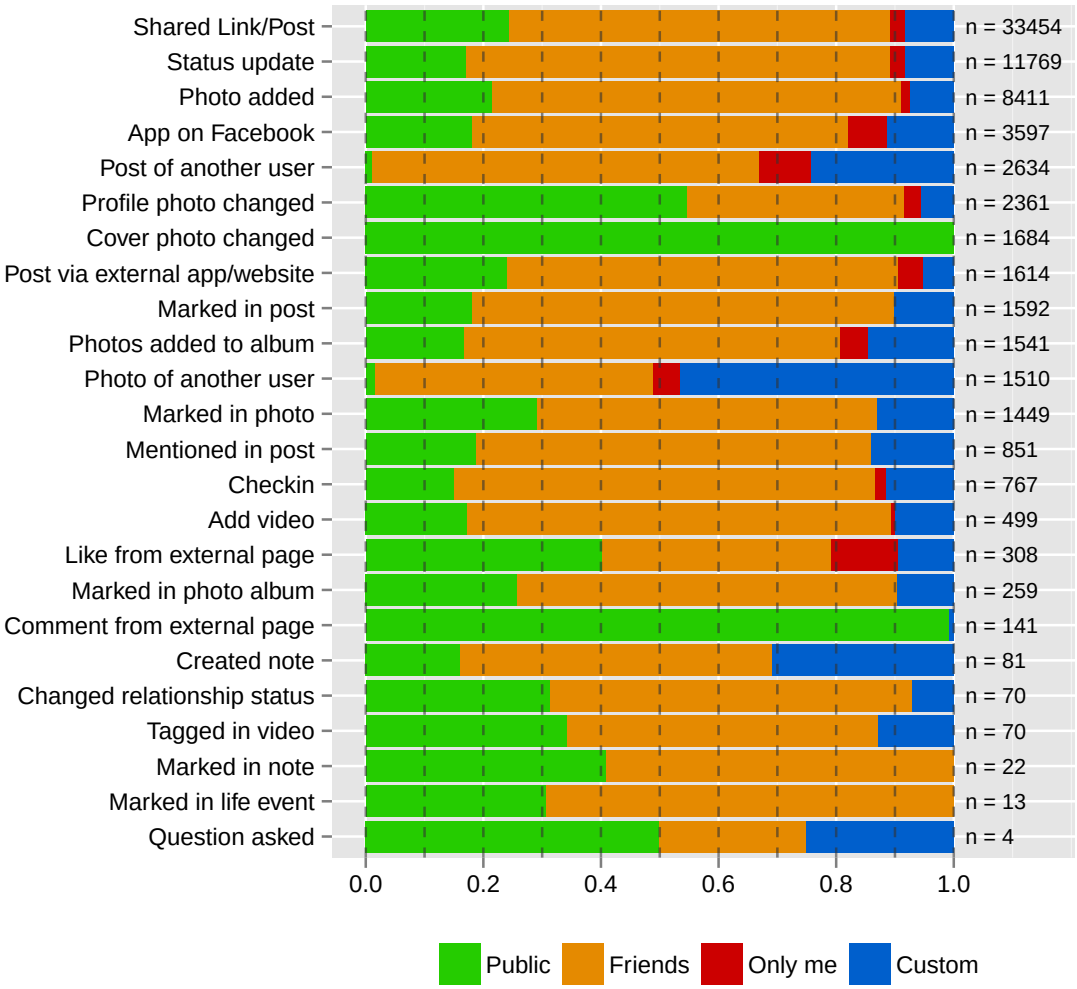


Figure 3.10: Privacy settings of timeline entries; Profiles may contain none or many timeline entries of each type; n refers to the total number of observed timeline entries per entry type

Privacy Impact of Simplified Audience Selection

Many Facebook users are unable to handle the privacy settings to meet their own sharing desiderata [Liu et al., 2011b, Madejski et al., 2011]. It is hence not sufficient to elaborate the actual privacy settings to study the sharing desiderata. Since the color-coding based privacy setting interface is shown to drastically decrease mistakes in selecting the audience [Paul et al., 2012c], elaborating the impact of the FPW helps to understand the gap between sharing interests and actual privacy settings.

With the help of our plug-in, 22.31% of the users change the visibility to a more restrictive setting, 19.55% of the users prefer less restrictive settings and 5.44% keep the average privacy by changing the visibility of different items equally to both directions. 52.14% of the users do not change the profile visibility compared to the settings before installing our plug-in.

The group of users who did not change any setting contains many inactive people with small user profiles as well as those who sent us feedback during the first session with activated FPW. All users who were not able to change any setting because of facing technical problems are also part of this group. In spite of not changing the settings, some users sent us feedback to state that the plug-in is very useful to check the settings with very little effort.

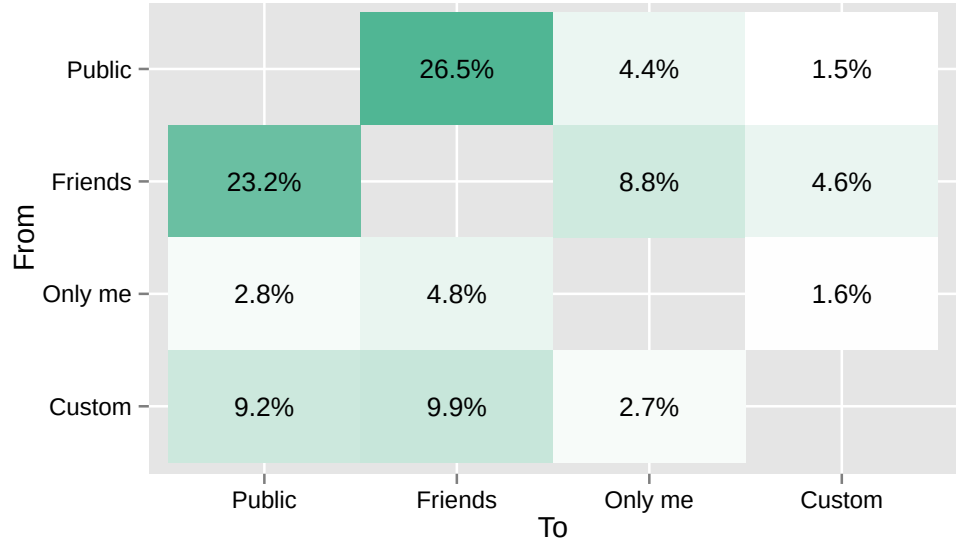


Figure 3.11: Heat map of visibility levels reflecting visibility change actions, performed with the help of the new interface (from, to)

In the remainder of this section, we focus on users who change the visibility of profile fields using the FPW. Figure 3.11 shows a heat map that illustrates change actions with respect to the visibility level before and after performing the actions. The most frequently performed action is to change the visibility from 'public' to 'friends'. The opposite change action is the second most frequently performed action.

With the help of the FPW, users hide more information ('only me') from public or friends than providing access to content. Remarkable is that the custom visibility setting, which is explicitly supported by our interface, is more likely to be removed than being newly used. Many users seem not to be happy to distinguish among different groups of friends. They instead prefer to either publish content without restrictions or among all friends.

Figure 3.12 depicts the exact percentage of items per profile field where users changed the visibility with the FPW. We only included those 2,816 users whose privacy has finally been affected by the FPW. The highest demand for changes can be seen in the timeline entries. A user profile in Facebook can enfold plenty of timeline entries but only a single entry in many other fields (e.g. birthday). The visibility of the employer has been changed by the second largest fraction of users, followed by the university and the friend list.

The tendency of performed changes towards more or less privacy in different profile fields is shown in Figure 3.13. Timeline entries, birthdays, about you, quotations, religious views, instant messagers, political views and e-mail addresses are those fields

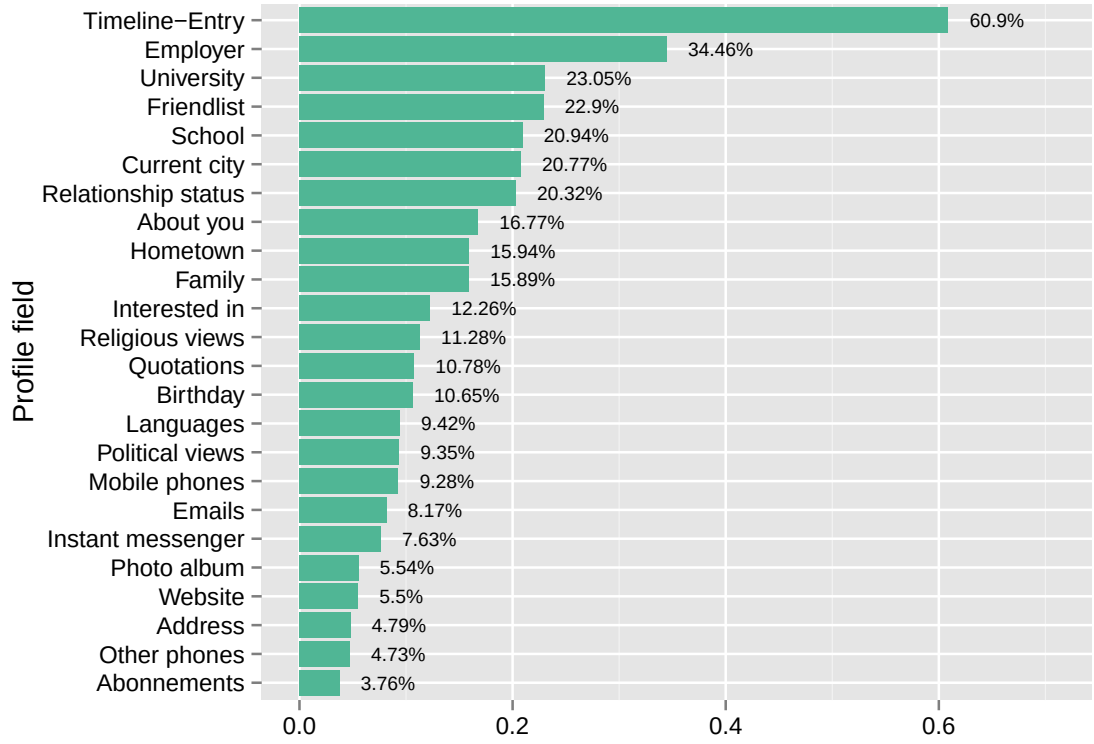


Figure 3.12: Percentage of users who changed the visibility of certain profile fields; only filled fields are mentioned

where more change actions towards less privacy have been performed. The rest of the profile fields are more private in average after using the FPW.

Comparison with Facebook Standard Privacy Settings

Advocates of the concept 'privacy by default' argue that people do not tend to change the default settings. Following this argumentation, and taking the user's audience selection efforts into account, an interesting question is how the defaults should look like to be in line with the user's needs. We thus compare the default settings with the actual privacy settings.

The Facebook default settings consist of two visibility levels: public and friends. The heat map in Figure 3.14 shows a comparison of the standard settings with the condition before applying the changes with the new interface: 43.6% of all profile fields, which are shared with public according to the Facebook standard, are publicly accessible. 39.2% of these public fields have been changed to be accessible only by friends. 49.2% of the by default friend-visible profile fields are still friend-visible before using the FPW and 38.4% of profile of the latter are visible to just a subset of friends.

Figure 3.15 illustrates the comparison of standard settings with the situation after using the FPW. In spite of many users changing profile settings, the cumulated amount of visible content does not change dramatically. 21.05% of the users used the plug-in to reduce the visibility of data objects in average by changing the standard settings. 10.44% changed the standard settings to the opposite direction. Our evaluation shows

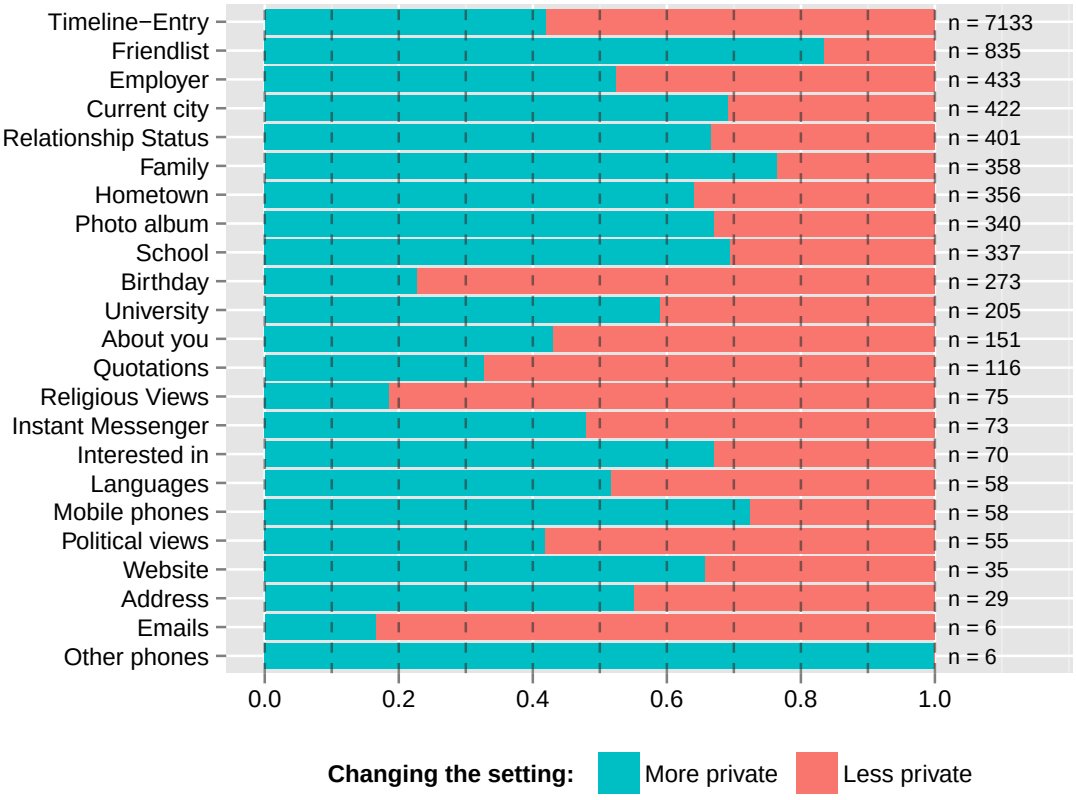


Figure 3.13: Fraction of change actions with the help of the FPW towards more or less privacy per profile field

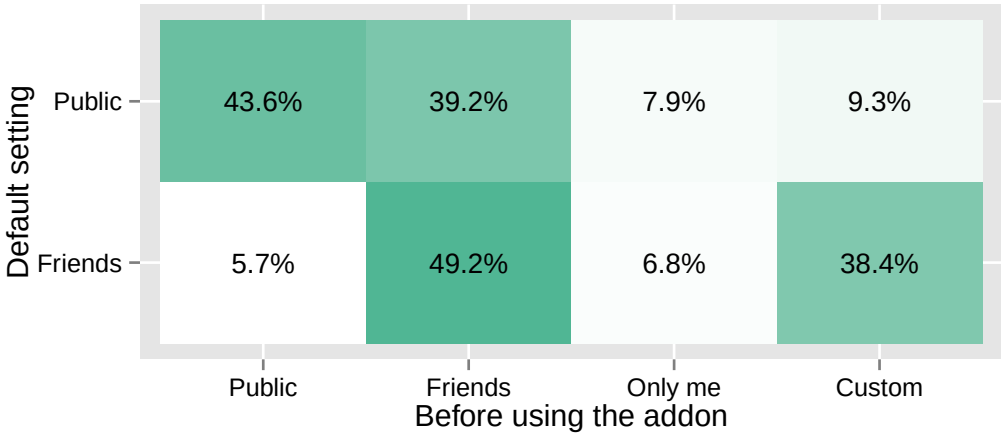


Figure 3.14: Comparison of Facebook standard visibility with the profile visibility before using the new interface

that the visibility of profile fields is still conform with the standard settings in many cases. 40.56% of the public fields are still unchanged after using the plug-in. That is also true for 49.13% of the fields which are friend-visible by default.

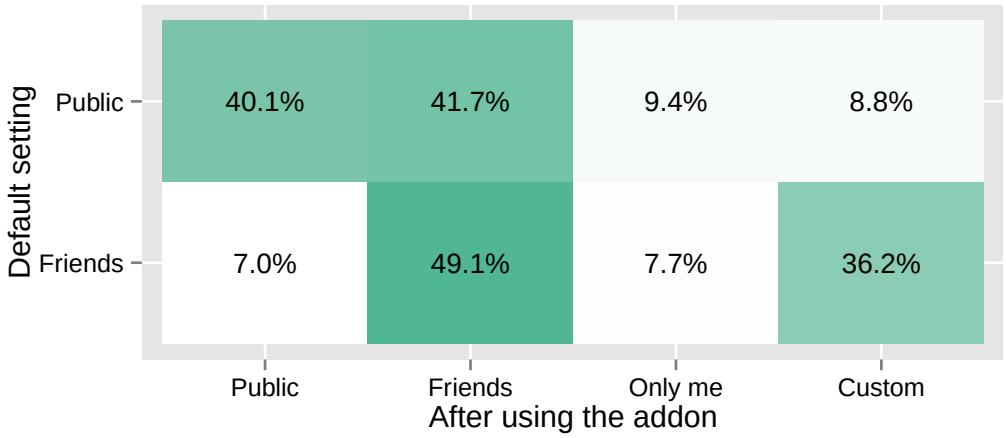


Figure 3.15: Heat map that illustrates the privacy setting changes from Facebook standard (ordinate) to individual settings (abscissa) after using the new interface

3.2.3 Country-Specific Privacy Evaluations

Since privacy preferences are depending on cultural backgrounds of users [John Rose and Christine Barton and Robert Souza and James Platt, 2014], we detail the global evaluations by comparing the actual privacy settings as well as the impact of the FPW with respect to the user's country of origin. Due to space limitations, we abstain from including every single profile field and concentrate on the examples showing the strongest variations.

As a result of constraints in our dataset, the cross-country comparisons suffer from differences in sample sizes. We address this issue in the following evaluations by normalizing all data and comparing only fractions (proportions) and medians which are rather stable with respect to different sample sizes. Also, we only include samples which are big enough to be stable against outliers and only apply extremely conservative statistic testing. Since we used the same method for acquiring study participants in all countries, we assume a potential bias to equally occur amongst the considered countries. Hence, we assume the comparability of our samples from various countries to be valid. Germany is a special case since our university is well known and receives more attention and trust here.

Exposure and Visibility of Personal Data in Different Countries

FPW users from different countries have different sharing interests. This can be shown by comparing both the information which is enclosed into the user profiles (filled fields) as well as privacy settings. Figure 3.16 shows the cumulated differences among the eight countries with feedback of more than 50 users. We cumulate all profile fields of all users in the respective country and compare the total proportions of content according to their visibility.

The most obvious result in our evaluation is that Egyptian users tend to share more information with the public than others. The latter also tend to hide the highest frac-

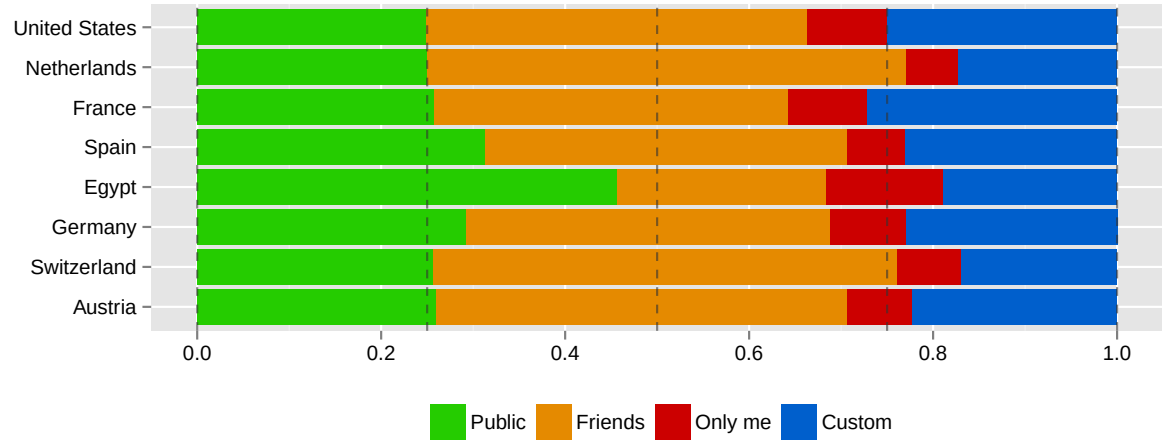


Figure 3.16: Cumulated privacy settings in different countries; sample sizes can be found in Table 1

Country	Country	W	p-value	BH	Setting
Egypt	Austria	3926	0.00010	0.00131	Friends
Egypt	Switzerland	1613	0.00015	0.00131	Public
Egypt	Switzerland	2380.5	0.00010	0.00131	Friends
Egypt	France	1974.5	0.00080	0.00378	Public
Egypt	Netherlands	1800.5	0.00039	0.00221	Public
Egypt	Netherlands	484	0.00018	0.00131	Friends
Germany	USA	29431	0.00779	0.02726	Only me
France	Switzerland	752.5	0.00533	0.02133	Custom

Table 3.5: Subset of significant results of the Pairwise Mann–Whitney U test of cumulated the data in Figure 3.16; W = test statistic; BH = Benjamini & Hochberg correction for multiple comparisons

tion of information (setting: 'only me') from anybody. Compared with the other seven countries, they tend to either publish content or not, rather than sharing with friends. We thus formulate the hypothesis that people in Egypt tend to use their Facebook profile as a tool to present themselves rather than to share content with their friends. Users from other Arabic countries seem to show a similar behavior, but the sample size is too small to provide meaningful results to include them into this study.

French users include the highest fraction of content to their profiles which is visible for just a subset of their friends. FPW users from Germany and the USA show significant differences in hiding content from others (setting: 'only me'). Many other differences can be seen (Figure 3.16), but they are not significant according to our extremely strict criteria.

We tested the significance of country-specific differences by applying the Mann–Whitney U-test (with continuity correction) on four distinct datasets. We compared (country pair-wise on user granularity) the country-specific percentages of the user profile field visibility to be either 'public', 'only friends', 'only me' or 'custom'. The Benjamini & Hochberg correction [Benjamini and Hochberg, 1995] has been applied to adjust p-values for multiple comparisons (28 pairwise comparisons). Table 3.5 provides the results.

Country	Country	W	p-value	BH	Field
Egypt	Austria	213.5	0.00064	0.01352	Languages
Egypt	France	10.5	0.00702	0.01776	Languages
Egypt	Germany	4613	0.00324	0.01469	Languages
Egypt	Netherl.	10.5	0.00248	0.01469	Languages
Egypt	USA	17.5	0.00154	0.0143	Languages
Spain	Austria	289	0.00539	0.0151	Languages
Spain	USA	53	0.00970	0.0209	Languages
Egypt	Netherl.	343.5	0.00097	0.01352	Hometown
Egypt	Germany	20119	0.00357	0.01469	Religious V.
France	Egypt	399.5	0.00407	0.01469	Family
France	Germany	22835	0.00761	0.01776	Family
France	Netherl.	495	0.00499	0.01508	Family
France	Switzerl.	316.5	0.00420	0.01469	Family

Table 3.6: Subset of significant results of the Pairwise Mann–Whitney U test of non-cumulated data; W = test statistic; BH = Benjamini & Hochberg correction for multiple comparisons

Country-specific content sharing differences can be even stronger realized by comparing the visibility of certain profile fields in different countries. We thus choose a sample of seven fields to explain the differences in Figures 3.17 till 3.23.

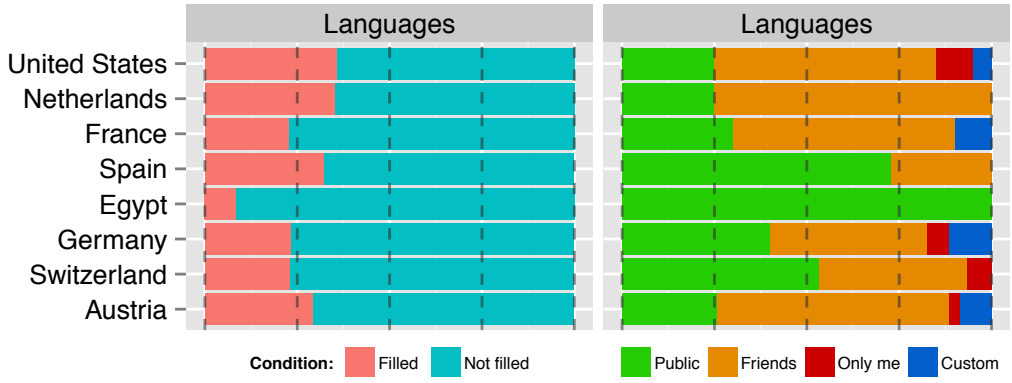


Figure 3.17: Privacy settings of the field 'languages'

Evaluating the languages field (Figure 3.17), we realized that Egyptian users do only rarely include the languages into their profiles. However, in case they do, they share this information with the public. This is a very different behavior, compared to other countries. We would thus suspect Egyptians not to speak other languages very often but in case they do, they seem to be very proud of it. Spanish users do share the information about their languages significantly more often than users from USA and Austria. This is less significant but still valid for Swiss users, too.

The profile field 'Mobile phones' is very special (Figure 3.18). Almost half of our Egyptian users included a phone number into their own profile. In contrast, Spanish FPW users show a three times smaller likelihood to add a phone number to the profile. However, the most interesting fact about this is that almost all users limited the visibility to just a subset of friends. That means that they need to change the default profile settings of Facebook when uploading this information to Facebook. The concept of privacy by design builds on the assumption that users do not tend to change default

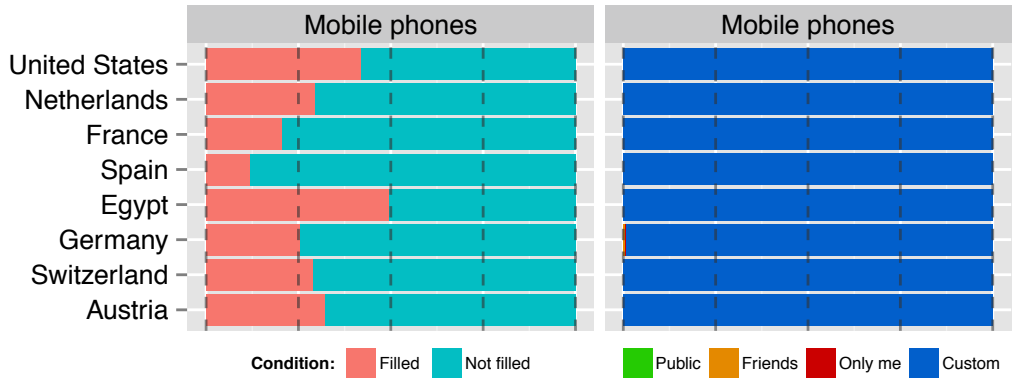


Figure 3.18: Privacy settings of the field 'mobile phones'

settings [Gross and Acquisti, 2005]. However, our observation is that nearly all users who entered a phone number, limited the access to the latter to a subset of friends. Our study hence does not support the assumption that Facebook users do not change default settings.

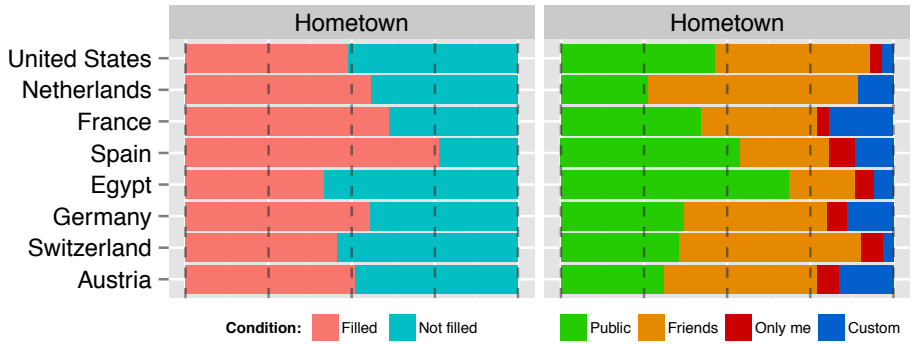


Figure 3.19: Privacy settings of the field 'hometown'

Another country-specific difference in sharing interest can be observed at the profile field 'Hometown' (Figure 3.19). Egyptian FPW users share the name of the hometown with a significantly higher probability with the public than FPW users from the Netherlands. However, the highest fraction of users who added the hometown to the user profile is from Spain.

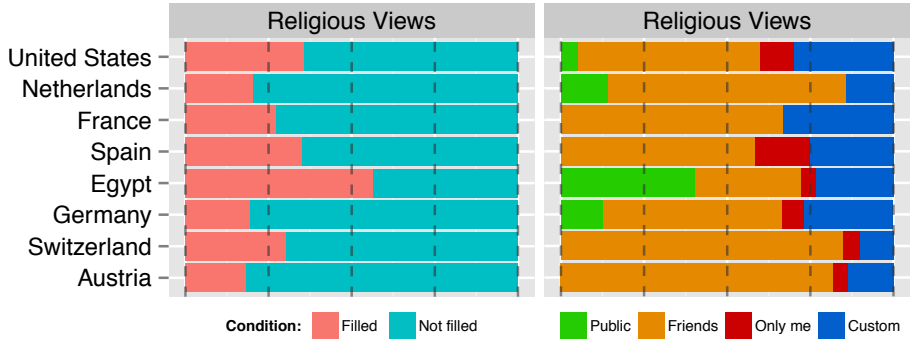


Figure 3.20: Privacy settings of the field 'religious views'

The religious views (Figure 3.20) are less likely to be included in the Facebook profile of the FPW users than e.g. the hometown or the family status. Only among Egyptian users, a majority of people can be observed to add the religious views to the user profile in Facebook. Furthermore, the Egyptians form the group that publishes this information with the highest likelihood. This observation can be used to found the hypothesis that religious views and their public commitments are more important in Egypt than in the other countries that we consider in this study.

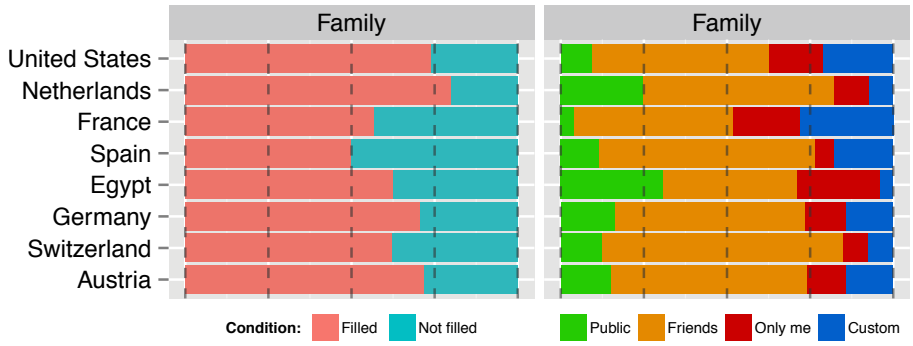


Figure 3.21: Privacy settings of the field 'family'

Information about the family status (Figure 3.21) is very likely to be included into the profiles. The overwhelming fraction of users prefer to share this information only with friends. In comparison to others, French users tend to restrict access to this profile field. Remarkable is that this is the field which is hidden by the largest fraction of people.

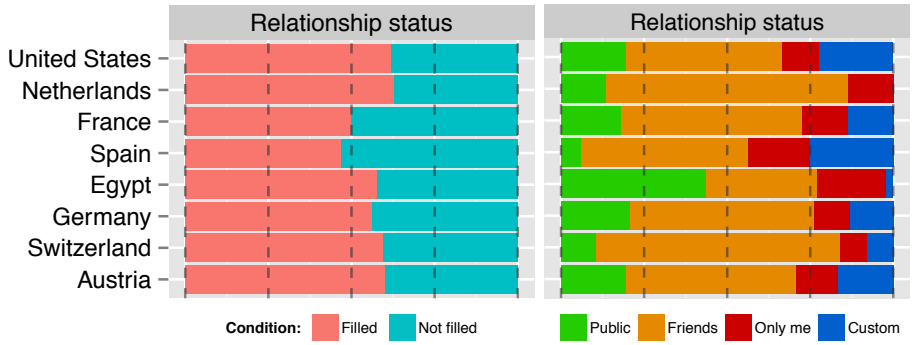


Figure 3.22: Privacy settings of the field 'relationship status'

Comparing the visibility of the relationship status of Spanish and Egyptian FPW users (Figure 3.22) is very interesting. Spanish FPW users are the subset with the lowest probability of filling and publishing the field 'relationship status'. With the highest probability compared to others, they share this information with only a selected subset of friends. In contrast, nearly half of the Egyptians publish their relationship status. At the same time, they are also the subset of FPW user with the highest likelihood to hide this bit of information.

The friend list (Figure 3.23) is the sole profile field in this evaluation which exists in every user profile without being empty. Users do not have the choice to upload a friend list or not: it is created automatically by adding friends. In case that users prefer not

to share this information, their only chance is to hide the list by choosing the visibility setting 'only me'. Accordingly, the latter setting is very popular. This is especially true for the subset of Egyptian FPW users.

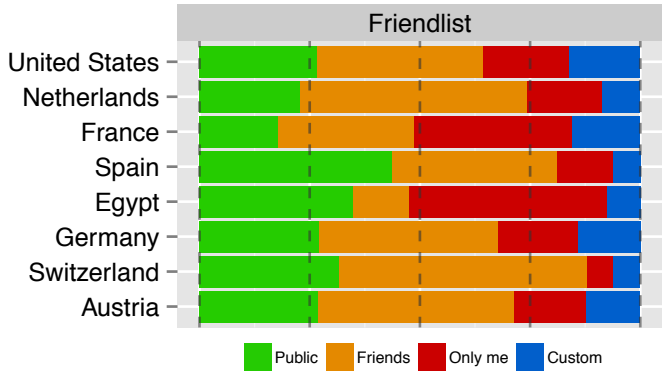


Figure 3.23: Privacy settings of the field 'friend list'

Country-Specific Changes of Privacy Settings

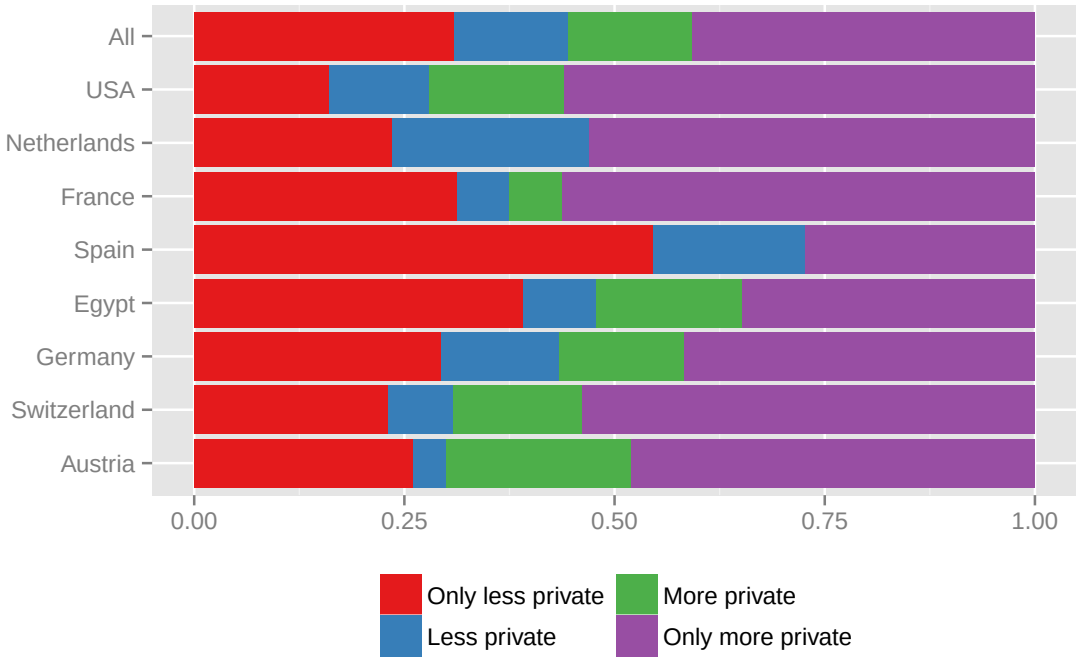


Figure 3.24: Fractions of users grouped by change directions of actions with FPW

In Section 3.2.3, we elaborated the privacy settings in different countries and distinguished between different fields. The main finding was that users from different countries share different information with their friends or the public. Since all users are faced with the same default privacy settings while having different sharing desires, the necessity of changing the visibility settings to meet the own sharing desires thus also differs. In this section, we elaborate the change actions which have been performed with the help of the FPW with respect to the user's country of origin.

Cluster	Friends		Likes		Photos		Map entries		Notes	
	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$
Only more private	171.48	112	106.03	39	22.34	3	38.69	4	0.48	0
More private	163.64	107	131.69	49	19.16	3	46.95	4	19.87	0
Less private	177.11	88	142.58	49	28.81	2	49.87	7	1.86	0
Only less private	186.49	91	125.69	55	22.69	4	98.32	11	0.71	0

Table 3.7: Profile statistic comparison with respect to change direction clusters; mean and median

In the remainder of this section, we distinguish among four subsets of users. The first subset, denoted *only less private*, consists of users who only changed visibility settings towards a higher visibility, e.g. from 'friends' to 'public' or from 'only me' to 'friends'. The second subset consists of users who changed the visibility *less private*. That means the users perform changes in both directions but those changes which grant more access to profile fields prevail the others. Accordingly, we denote the third and the fourth subset *more private* and *only more private* were the third subset consists of users who mainly changed to a more private setting and the fourth subset of users who only restricted access to profile fields.

We ignored two subsets which could be built when following the previous logic: those users who did not change anything and those users who changed the privacy settings equally to both directions. The latter have been ignored since the subset contains many users who only tried our new interface and changed one field in both directions. The subset of users who did not change anything can hardly be evaluated since this subset contains those users who faced technical problems, thus unable to perform changes.

Figure 3.24 shows the distribution of the four clusters in our top eight countries. The relative cluster sizes are different amongst the mentioned countries and the majority of the FPW users change the visibility of profile fields towards one of the two possible directions. Surprisingly, in spite of advertising the FPW as a tool to increase the privacy, the fraction of users who only used the FPW to change the privacy settings to less private settings is relatively high (30.92% of the sum of the four clusters). In Spain, the latter is even higher than 50%. In total, the FPW caused less information to be accessible.

Comparing the privacy settings in Figure 3.16 and the change actions in Figure 3.24 draws a homogeneous picture: The two countries with the least conservative settings are those with the highest fraction of users in the cluster *only less private*. Switzerland and the Netherlands are at the opposite of the range in both illustrations.

3.2.4 Change Direction Clusters

The clear distinction of clusters in Figure 3.24 inspired us to evaluate the differences in the user profiles to examine implications of privacy desiderata on profile properties.

Cluster	Friends		Likes		Photos		Map entries		Notes	
	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$	$\varnothing_X$	$\tilde{X}$
United States	201.41	115	156.47	73	45.64	8	24.68	4	1.93	0
Netherlands	111.63	90	33.50	10	17.70	8	47.73	21	0.03	0
France	267.81	112	129.22	44	47.78	10	82.63	5	30.86	0
Spain	148.78	117	127.97	23	98.37	60	102.05	7	2.94	0
Egypt	331.86	150	455.70	181	61.86	17	25.29	2	7.93	8
Germany	154.54	93	102.22	35	15.38	2	45.61	4	0.52	0
Switzerland	225.52	131	185.41	40	28.30	6	67.48	9	50.58	0
Austria	275.83	193	171.46	96	36.55	16	108.91	5	1.60	0

Table 3.8: Profile statistic comparison with respect to countries

In Tables 3.7 and 3.8, we thus compare the mean and median of the countable profile properties 'Friends', 'Likes', 'Photos', 'Map Entries' and 'Notes' with respect to clusters and countries.

Users in the cluster *only more private* have more friends (median) than others but less likes and less map entries. Users in the cluster *more private* have still more friends than those who used the FPW to increase the visibility of profile fields. Also notable is that users in the cluster *only less private* do not mind to tell Facebook their location by having more map entries. Notes are not very popular amongst our set of users. The mean of 19.87 in the *more private* cluster is a result of a few freak users having plenty of notes.

Table 3.8 shows the mean and the median of the same set of countable profile properties as they can be found in Table 3.7. Obvious differences between country clusters are that Egyptian FPW users who sent us feedback have more friends and more likes than all others. The cluster of Dutch FPW users is the opposite extreme, having 18 times less likes (median) than Egyptian cluster. The Spanish users share 60 pictures while the German users share two (median).

Comparing the differences amongst our four change direction clusters in Table 3.7 exhibits notably smaller differences than comparing user profile differences amongst users from different countries in Table 3.8. All values in Table 3.7 are very close to the values in the line 'Germany' in Table 3.8. The reason is that the majority of the FPW users in this study are Germans. It underlines the influence factor *country of origin* to dominate the *change direction*.

3.2.5 Related Work

Privacy is a topic that is broadly addressed by plenty of publications in computer science. In this section, we discuss works on privacy in OSNs with the focus on user behavior and interface construction rather than systems or algorithms. Since we discuss a new privacy settings interface, default privacy settings and privacy awareness in this section, we particularly focus on publications on privacy by design as well as on publications, suggesting interfaces for privacy settings in OSNs.

Works on privacy by design are built on the assumption that people do not tend to change their privacy settings. Gross and Acquisti state that "We can conclude that only a vanishingly small number of users change the (permissive) default privacy preferences" [Gross and Acquisti, 2005]. Based on this logic, the authors suggest to implement default privacy rules that prevent leakage of data. In contrast to this study, we evaluate how much a better interface helps the users to meet their needs by avoiding misconfiguration, and compare the sharing desiderata with respect to the user's country of origin. Furthermore, our results show that more than 59% of the privacy settings do not stay untouched in case of using our plug-in.

In 2008, Krishnamurthy and Wills [Krishnamurthy and Wills, 2008] examined privacy settings in Facebook, Myspace, Bebo and Twitter based on crawler-gathered data. They discovered that there is some use of privacy settings but there is still a significant portion of users who allow strangers to access private information. They further examined the amount of information that is shared within regional networks and discovered a negative correlation between network size and the amount of shared information. In comparison to [Krishnamurthy and Wills, 2008], we focus on Facebook, obtain our data directly from the uses, evaluate the impact of our color-based privacy setting interface and get different results regarding the users disposition to change privacy settings.

Stutzman et al. [Stutzman et al., 2013] monitored the public-available data of 5,076 members of the Carnegie Mellon University from 2005 till 2011. They discovered an increasing privacy awareness over time. Johnson et al. [Johnson et al., 2012] surveyed 260 participants from the United States, recruited via ResearchMatch, by using a Facebook application. They asked questions with the background knowledge which was obtained by reading the participant's Facebook profile via API. Inter alia, they discovered that 94.6% of their participants denied access to their content by people outside their friend network. Mondal et al. [Mondal et al., 2014] studied the use of social access control lists (SACLs). The friend list usage of 1,165 users of the tool "Friendlist Manager", has been analyzed. They found "that a surprisingly large fraction (17.6%) of content is shared with SACLs. However, we also find that the SACL membership shows little correlation with either profile information or social network links; as a result, it is difficult to predict the subset of a user's friends likely to appear in a SACL."

Beside the FPW, other approaches to help users to mitigate the misconfiguration exist, too. Lipford et al. [Lipford et al., 2008] suggest to allow users to take the point of view of the expected audience. PViz [Mazzia et al., 2012] is a privacy setting approach based on group visualizations in different granularities. Carminati et al. [Carminati et al., 2006] suggest rule-based privacy settings that define types of relationships and a set of rules which type of relationship is a precondition to access a certain data object. Fang et al. [Fang et al., 2010] propose a machine learning based approach which implements a wizard that suggests a set of access rules. The idea is to learn implicit rules which are applied by users to set the visibility of objects. In contrast, our interface allows both to quickly grasp the visibility of content items based on a color coding and to change those settings with a single click.

Other related work can be found in studies about Facebook user statistics[11], a report[12] about the evolution of privacy in Facebook and a survey in [John Rose and Christine Barton and Robert Souza and James Platt, 2014] where consumers have been asked which information they consider to be private. However, the user statistics do not provide information about privacy settings and the consumer survey does relies on questionnaires without a concrete link to social networks.

3.2.6 Summary and Conclusion

In this chapter, we presented the first large-scale study about content sharing and privacy preferences of Facebook users with special focus on country-specific characteristics. It is based on 9,292 feedbacks from 4,182 users in 102 countries. Our sample is neither complete nor a result of a random sampling process (Section 3.2.1). Yet, the huge media attention from radio stations and daily newspapers, which address ordinary people, shows that the FPW was assumed to be interesting for their recipients. Furthermore, the fact that a very big fraction of users discloses more information instead of hiding it with the FPW is a strong evidence that it is not used by a fringe group of privacy savvy people.

In contrast to related work in the field of privacy preferences, we collected our data on the users' clients and evaluate the behavior from real users who perform audience selection on their own user profiles for their own reasons. However, even the evaluation of the actual privacy settings is only a rough estimation of the sharing preferences that suffers from two imprecisions: (i) Many users are unable to properly choose their audience with Facebook's privacy setting interface, and (ii) the sharing preferences exhibit a vast diversity depending on the user's country of origin.

To overcome those imprecisions, we evaluated changes that have been made using color-coding based privacy controls. In a previous study, the latter have been demonstrated to be usable, intuitive and effective to drastically reduce errors and efforts in selecting the audience [Paul et al., 2012c].

We further elaborated the country-specific differences in both the privacy settings as well as the privacy change actions. Additionally, a cluster analysis highlighted the relation between the impact of the FPW on users' audience selection decisions and their countable profile properties.

When creating an account in Facebook, it is obligatory to reveal information about gender, e-mail and birthday. However, our results indicate that the majority of FPW users sufficiently trusts Facebook to confide personal information such as family status, current city, hometown, employer and school. Contrariwise, only a minority of FPW users includes information on skills, addresses or political views into their profiles.

The most popular audience selection strategy is to allow all friends to access a certain bit of information, followed by publishing it and disclosing it to only a subset of

[11] `http://blog.stephenwolfram.com/2013/04/data-science-of-the-facebook-world/`, accessed on 2015-10-25

[12] http://mattmckeon.com/facebook-privacy/, accessed on 2015-10-25

friends. The setting 'only me' is the least popular setting. Beside unpopular features such as subscriptions and websites, the current city, the hometown, languages and the employees are the most frequently published bits. Only very few FPW users publish their e-mail address, instant messenger ID and their birthday, but the majority shares these bits with their friends. The friend list is a divisive issue amongst users to decide about its audience. Being published by more than one third of all FPW users, the friend list is the profile field that the second largest fraction of users is hiding (setting 'only me').

Introducing the comprehensible color-coding interface of the FPW impacts the audience selection of users. In spite of the FPW being advertised as a privacy tool, users disclose selected bits of information to the public and to the complete set of friends. Users mainly change the privacy settings for timeline entries, the friend list and the profile field 'employer'. While the visibility of the timeline entries and the field employer are roughly equally switched to more and less restrictive privacy settings, the friend list setting was preferred to be more restrictive by 83% of our participants. The total amount of content that is visible to Facebook users does not dramatically decrease after introducing a comprehensible visualization of privacy controls, but the composition of the visible content changes. This indicates that the usability of Facebook's privacy setting interface can be improved by using color codings.

Which information is uploaded to Facebook as well as which information is shared with whom is strongly depending on the user's country of origin. A perspicuous example is that less than 22% of the German FPW users shared their religious views on Facebook while the majority of Egyptian FPW users included their religious views into their user profiles. The visibility is chosen accordingly. Thus, global default privacy settings cannot meet the sharing interests of all users since the sharing interests show country-specific as well as person-specific differences.

Authors of alternative OSN architectures argue that fine-grained access control is an important feature to improve privacy in OSNs [Jahid et al., 2011, Simpson, 2008, Carminati et al., 2009]. However, our FPW users tend to remove group settings and individual access rules to achieve a lower complexity of access rules. We construe this fact to express user's favor for simplicity and thus encourage privacy interface designers to focus on simplicity rather than on a rich set of functionality.

Chapter 4

Improving Privacy by Decentralizing OSNs

Simplifying the audience selection helps to avoid adverse effects which are caused by over-sharing. However, today's OSNs are each operated by a commercial provider who is the explicit authority in the respective OSN. Being the operator of the system, the latter has omnipotent power to access and monetize user data. Users are at the mercy of the provider's goodwill not to misuse their data.

In addition, applying steganography and encryption does not abolish privacy-related side effects. While using steganography is no feasible solution to hide frequent communication and large items such as photos or videos, the OSN provider can still learn valuable information about users in case of using cryptography by evaluating messages (or ciphertext). Some examples for information that may be inferred even when applying cryptography are:

- **Service usage patterns (Churn):** How intensive do users use the service? How do users' diurnal habits look like?
- **Communication partners:** Who talks to whom?
- **Communication intensity:** How frequent do OSN users communicate? At which time of the day? The nature of a relationship between communication parties can potentially be guessed.
- **Type of content:** The technical size of content items allows to identify e.g. pictures or videos with a very high probability.
- **Technical equipment:** By understanding picture encoding, technical properties such as picture sizes allow for guessing devices.

Economic pressure to earn money due to provider-side infrastructure and maintenance costs and the provider's legitimate profit interests lead to strong incentives for OSN providers to monetize user data far beyond the user's sharing interests [Falch et al., 2009]. It is unclear whether content encryption still allows OSN providers to keep their advertisement-based business models or to find other bearing alternatives.

The importance of OSNs for the daily inter-person communication puts the OSN providers in a position of being gate keepers to parts of the social life of their users. Forced by this dependency, users strongly tend to accept side effects and even disadvantageous terms of usage, since the OSN providers may exclude users from the OSNs and subsequently from parts of their social contacts. One example is that users are forced to grant usage rights to Facebook: "For content that is covered by intellectual property rights, like photos and videos (IP content), you specifically give us the following permission, subject to your privacy and application settings: you grant us a non-exclusive, transferable, sub-licensable, royalty-free, worldwide license to use any IP content that you post on or in connection with Facebook (IP License)."[1]

Authors of decentralized OSN (DOSN) approaches aim to abolish OSN providers and the side-effects of centralized OSNs by creating decentralized systems, providing the social networking functionality. Since no single authority controls the whole network in DOSN, nobody has access to all user data. Many kinds of DOSNs have been proposed by several authors. Nevertheless, the idea of decentralizing OSNs has not been widely adopted. Besides Diaspora[2], none of the DOSNs has a noteworthy user basis. In contrast to the authors of many DOSNs, Narayanan et al. doubt in [Narayanan et al., 2012] that decentralizing OSNs is a feasible way to build social networking services. We argue that decentralizing OSNs is a worthwhile idea and aim to help the DOSN community with this survey by elaborating and evaluating what has been suggested in the field of DOSN.

In this chapter, we explain the concept of decentralizing OSNs, survey the state of the art and elaborate the remaining challenges in the field of DOSN.

4.1 Requirements and Adversary Models

In this section, we discuss the requirements that are our benchmark to evaluate DOSNs and introduce the adversary models that are later used to discuss the security of DOSNs.

4.1.1 Requirements

In the remainder of this section, we present and discuss different DOSN approaches, but none of them has a deployment with a reasonable user base and the full set of functionality compared to today's most popular OSN, Facebook. We consider the non-academic approach, Diaspora, with about 400,000 users[3] to be the most successful DOSN. It still comes without a recommender system for friends and content and without a system-wide content and profile discovery mechanism.

Elaborating success determinants of DOSN is out of the scope of this survey, but following [Narayanan et al., 2012], we assume that it is a necessary success-precondition for DOSN to implement attractive functionality in a usable way. Subsequently, we

1 https://www.facebook.com/terms.php, accessed on 2015-04-09
2 https://joindiaspora.com/, accessed on 2015-04-09
3 https://diasp.eu/stats.html, accessed on 2015-04-09

assume that users do not completely trust their OSN providers not to misuse private data [Dwyer et al., 2007], but that they do not want to abdicate benefit of OSN functionality. Hence, DOSNs need to become as usable and as useful as their centralized counterparts in addition to respect user's privacy to become successful competitors.

In our discussion we thus use Facebook as a baseline for the Quality of Service (QoS). Since we do not have access to implementations of all proposed DOSN systems (the majority of approaches are scientific and thus implementations may even not exist), we discuss the approaches subsequently represented by the set of functionality and performance properties of the proposed DOSN. Furthermore, censorship resistance, security issues and economical issues are discussed in the remainder of this survey. We thus assume that DOSNs need to provide a comparable level (compared to centralized OSNs) of service quality while being better in terms of privacy in order to be considered as an alternative with respect to:

1. System performance
    - message transfer and profile update delays
2. Privacy of content and interactions
    - confidentiality and integrity of communication
    - user authentication and access control
    - accountability of user actions within the system
    - incidential data evaluation vulnerabilities (e.g. in case of cipher text access)
    - resistance to censorship
3. Functionality
    - user handle and content search functionality
    - recommender systems
    - API
4. Economical issues
    - network infrastructure costs and storage resource provisioning
    - type of payment (e.g. money to rent servers or resource contribution via a P2P approach)

4.1.2 Adversary Models

The existence of an omnipotent Social Network Provider (SNP) is considered to be a privacy problem by the authors of DOSN approaches. The underlying assumption is that the provider can neither be trusted to protect user data from external attackers nor to withstand misusing the data for monetization purposes. However, the OSN provider maintains a closed system with little attack surface for external attackers. The question thus is whether decentralization is the way to go for improving privacy. To discuss this issue, we define the following set of attackers:

1. An adversary with read and write access to all data, stored in the system (curious omnipotent SNP).

2. A traffic observer, having an Internet Service Provider (ISP)-like view at the network traffic.
3. An adversary who can enforce all authorities (organizations and companies like SNP and ISP) to corporate with her (governmental attacker).
4. The mass data collector, collecting as much data about as many users as possible (e.g. crawler).
5. The stranger adversary who represents an arbitrary user of the OSN (no direct friendship connection to the attack target).
6. The friend adversary (defined in [Greschbach and Buchegger, 2012]), exploiting the friend connection in the OSN.
7. The online reputation attacker, aiming at destroying the reputation of individual users (cyber bullying).

4.2 DOSN Architecture Model

The following 3-layer DOSN - architecture model (Figure 4.1) introduces an abstraction of the DOSN design space. Subsequently, it describes its components which are addressed by approaches, covered in this survey or are core functionalities of today's popular OSNs. Existing DOSN approaches individually take just a subset of the optional extensions into account, but minimally specify the DOSN – core layer.

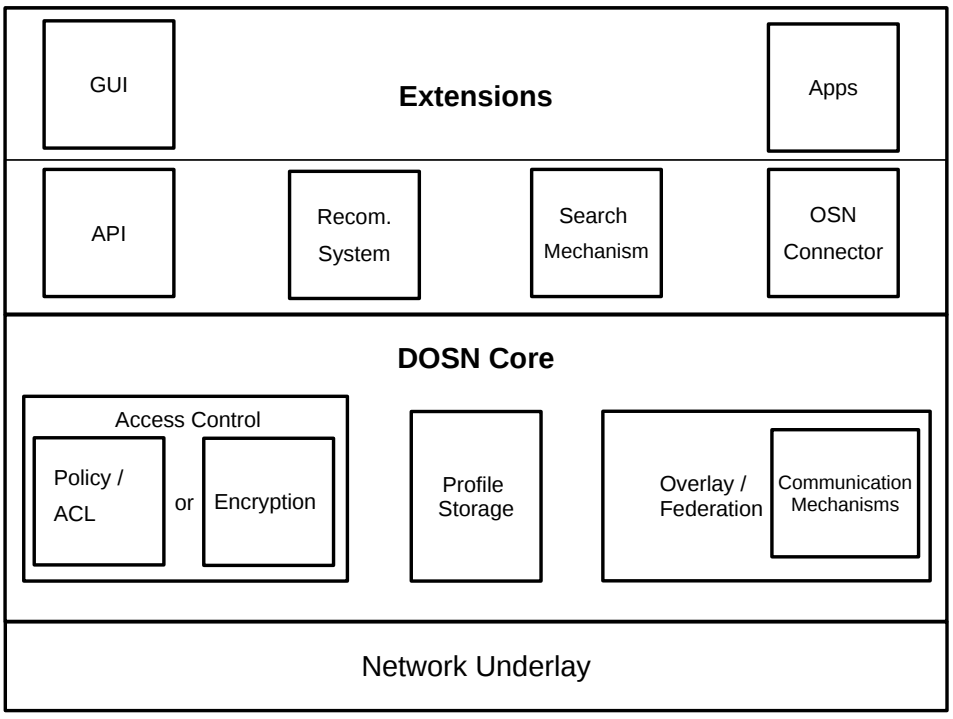


Figure 4.1: DOSN architecture model

We derived this model by examining the technological components of DOSNs that are necessary to implement the intended functionality. Each atomic model component represents a class of algorithms or software design patterns. The authors of DOSN ar-

chitectures thus determine the properties of their DOSNs either by combining existing mechanisms or by developing new mechanisms to implement the respective functionality.

The lowest layer represents the communication network which is used for all participating entities to communicate. We assume that it reliably transmits messages from one entity to the other. The middle layer, called "DOSN - core" contains all components which are necessary to provide the basic DOSN functionality. The upper layer represents extensions aiming at making the user experience enjoyable. It is divided into two sublayers where the lower part is hidden from the users and thus provides services for the elements of the upper part, facing direct user interactions.

The core component contains three main parts:

- the access control component which can be realized via access policies, encryption schemes or a combination of both,
- the profile storage component which describes how profile data is stored in the system and
- an overlay or federation component to organize the communication among nodes. We distinguish among protocols supporting direct user interactions (communication) and those supporting technical information exchange (e.g. profile update propagation) where the latter is transparent to the user and hence may raise different time and volume requirements.

The extensions layer consists of two sublayers. Only the components of the upper layer face direct user interactions while components of the lower sublayer are hidden from the user. We define the following hidden modules of the extensions that implement the extended functionality

- API as an interface for third party applications
- a recommender system which can potentially recommend both: friends relationships to create as well as content items to consume
- a search scheme which supports privacy preserving search for user handles or content addresses
- a social network connector, connecting the new DOSN to existing OSNs, since network effects yield the largest OSN the most attractive. The probability of finding friends is growing with a growing number of users.

The two components with the closest position from user perspective (and hence not hidden from the users) are the GUI (graphical user interface) and the applications which can be built by third parties or users. We consider both components to be on the same layer since applications may realize own GUIs.

4.3 Design Decisions

Decentralizing OSNs seeks to implement OSN functionality without relying on an omnipotent service provider for resource provision. DOSNs thus need to implement a mechanism to satisfy storage resource requirements to support user profiles. For the sake of satisfying users' privacy needs, it is necessary to support mechanisms that allow

users to restrict information access to a chosen audience. Since OSNs are (by definition) communication systems, the minimal DOSN setting also comprehends some kind of communication mechanism.

The design decisions, which are necessary to implement this minimal setting, need to be made in every DOSN approach. They are the foundation for our three main classification criteria: i) the way of decentralizing the storage of content, ii) the mechanisms to decentralize access control and iii) the way how decentralized interaction and signaling mechanisms are implemented.

4.3.1 Decentralized Storage

Decentralized storage of user-owned data is a very strong concern of the authors of OSN approaches. The main idea is to put the data owner into a position to hold sway over her PII by keeping the data storage in her influence zone (sphere wherein the user or a trusted third party, e.g. a friend, is directly capable to determine storing, erasure and access operations). In fact, different storage concepts strongly characterize the different DOSN approaches, since they have a big impact on the nature of the architecture itself. Three different fundamental types of decentralized storage of content have been proposed: storing on i) peer nodes (P2P-OSN), on ii) external permanent servers (F-OSN) or on iii) a mixture of both (Hybrid OSN).

In the case of *storage on peers* (users' devices), one main challenge is to handle resource unreliably. To minimize the risk for data loss and data unavailability, redundant service provision is mandatory. Different performance implications of redundancy procuring approaches yield this design decision crucial for P2P-OSNs. Since replication of resources is the only type of redundancy leveraged in the literature, one important storage-related system design decision for P2P-DOSN approaches is to choose the nodes where to save the copies (i.e. replica placement).

The suggested replica placement strategies are meant to store data:

- at random locations in the network
- on a set of strangers' devices
- on friends' nodes
- on a chosen subset of friends
- by leveraging a DTH

To circumvent the availability issue, the replica placement and maintaining effort, storing on reliable *external servers* has been suggested. Profile owners in DOSNs have - in contrast to centralized OSNs - the choice where to store their data. The criteria for this decision are monetary cost, trust toward the service provider or guaranteed levels of availability and reliability. We distinguish between flat storage on arbitrary resources and F-OSN based on specialized servers implementing OSN logic at the same place.

Hybrid approaches may allow both: storing data on dedicated servers as well as storing locally on churn-exposed peers.

4.3.2 Decentralized Access Control

Since one goal of DOSNs is to improve privacy, they should allow users to define exactly who is part of the set of legitimate content recipients for each single piece of content. Two general primitives have been suggested: access control (AC) which is performed by trusted entities as well as encrypting content and distributing keys among legitimate recipients. Furthermore, mixtures of both primitives are part of some approaches.

Approaches relying on ACLs are based on the principle that users have to prove they own necessary rights to an authority enforcing policies defined in the ACLs to access or modify a given piece of content. In our classification, we characterized works relying on ACLs based on who enforces the policies, which can be peers or external servers (based on where data are stored). Finally the ACLs can be enforced by external services in the form of applications or plug-ins, run on the top of the platform, which allow users to interact with each others.

Data encryption approaches are based on the principle that *anybody can retrieve a piece of content, but only users who have decryption keys can interprete it.* Relying on content encryption for performing AC, implies the definition of a key management mechanism. In our classification, we thus characterize several works based on the adopted mechanism.

A motivation for implementing both: an ACL as well as an encryption scheme is that ACL does not protect from access to encrypted content (ciphertext) and thus still allows for inferring communication details like e.g. the data size or communication patterns [Greschbach et al., 2012].

4.3.3 Interaction and Signaling Mechanisms

Interactions among users in terms of sending messages are at the core of any social platform and may include signaling and notification and establishment of new relationships etc. In centralized OSNs, the service provider mediates interaction among users.

In DOSNs, interaction mechanisms can be either i) still centralized, meaning that they are handled by one single logical entity (in some cases DOSN still rely on classical centralized counterparts for handling interactions [Liu et al., 2011a]) or ii) decentralized. Decentralized interaction can be realized based on a P2P substrate, relying on direct interactions among user terminals, on publish-subscribe models or on federation protocols including inter-server communication (e.g. XMPP). Some DOSNs do not define how such mechanisms should be implemented, rather addressing lower level aspects and relying on higher level plug-ins for handling interactions.

In our classification, we thus distinguish between centralized and decentralized handling of interactions and point out the adopted approach.

Arch.	Ref	Degree	Storage		AC		Interact Mech.		Comments
			Peers	Server	ACLs	Encr.	Centr.	Dec.	
P2P-OSN	PeerSoN	FD	Previous download	-	-	- PKI	-	Direct + DHT	Support for direct interactions (also with no Internet access)
	Safebook	FD	Trusted friends	-	-	PKI	-	DHT	Anonymity of interactions via encryption and recursive hop-by-hop routing
	LifeSocial. KOM	FD	DHT	-	-	BE	-	Plugins	Interactions based on external applications (plugins)
	LotusNet	FD	DHT	-	-	PKI	-	DHT	Based on Likir
	DECENT	FD	Random nodes (DHT)	-	-	ABE	-	DHT	Social network functionality on top of EASIER
	Cachet	FD	Random nodes (DHT)	-	-	ABE	-	DHT	Performance improvement on DECENT
F-OSN	SoNet	FD	-	Active	Servers	OOB or SMP	-	XMPP	XMPP-like architecture / social graph obfuscation
	Mantle	FD	-	Passive	-	OOB	-	Pub/Sub model	Group encryption on any storage, pub/sub for interactions
	PrPl	FD	-	Active	Cloud buttler	Undef.	-	Plugins	Cloud buttler either at home or in the cloud / own language: SocialLite
	Diaspora	FD	-	Active	Hosting nodes	-	-	Hosting nodes	Trusted social hubs, hosting several user pods each
	[Anderson]	D	-	Active	-	PKI	-	Pub/Sub	Multi-layer clients with sandbox for external applications
Hybrid	Vis-a-Vis	FD	-	Passive	User pod	-	-	DHT	P2P substrate, data stored in user pods on personal devices / cloud services
	[Kryczka]	D	Social graph, locality	Active	Hosting node	-	Central Index	-	Centralized OSN extended with P2P content storage
	[Raji]	D	-	Active	-	BE	-	Pub/Sub	Private data on personal storage, rest at OSN provider
	Polaris	FD	-	Passive	User Home	-	-	Ext. apps + direct	Storage on phones or servers, NAT traversal necessary
	Confidant	D	Trusted friends	-	-	OOB	Extern. Platform	-	Storage on trusted servers, existing OSN is used for signaling (notification)
	Vegas	FD	-	Passive	-	PKI	-	direct	P2P/reliable storage

Table 4.1: DOSN approaches; (D = distributed, FD = fully distributed, BE = broadcast encryption, OOB = out of band, ABE = attribute based encryption)

4.4 Resulting Effects of Design Decisions

In this section, we discuss the properties and implications of the respected classes entailed by the basic design decision elaborated in Section 4.3.

4.4.1 Decentralized Storage

The answer to the question of where to store data in DOSNs naturally commemorates the impacts of the issues of data availability, storage costs as well as trust. Data availability is an important issue in case of using unreliable resources (P2P). Storage costs become important in case of applying replication schemes that maintain multiple copies in the network or in case of using dedicated resources. In either case, the storage devices have to be trusted to reliably serve legitimate requests and not to leak or misuse accidental data or even the content itself if it is not encrypted.

In P2P-OSNs [Buchegger et al., 2009b, Cutillo et al., 2009b, Graffi et al., 2008, Aiello and Ruffo, 2012, Jahid et al., 2012, Nilizadeh et al., 2012] data availability is bound to the on-line time of the different principals and can be enhanced thanks to the discussed replication mechanisms. In [Cha et al., 2007] several replication mechanisms are discussed, which show how availability increases with replication granularity.

No matter how the replication nodes are chosen, storage on peers costs storage as well as bandwidth resources which are not for free. Sophisticated approaches [Koll et al., 2013, Shahriar et al., 2013] aim at minimizing resource consumption while maximizing profile availability. We briefly describe them in Section 4.7, since these approaches are just replication schemes rather then complete DOSN approaches according to our definition and hence not part of our classification.

However, data replication may affect data consistency, since the latter is significantly harder to achieve as the number of copies of a single piece of content, distributed on several nodes, increases. From the user's point of view, all replication schemes, suggested by the authors of DOSNs, come with serious disadvantages:

1. Storing replicas at friends' nodes
    - Bootstrapping: it is difficult when entering the network while having no friends.
    - Correlated failure: the profile cannot be found by unconnected friends and strangers if all friends are offline at a given point in time.
    - Load balancing is not scalable to popularity peaks if the set of replica nodes is fixed and limited to the friend's devices, assuming that profile data items can be requested publicly (e.g. requests caused by a newspaper article about a person).
2. Random replication without management requires a to high number of replicas to be feasible under realistic churn assumptions. [Paul et al., 2012a]

3. Passive replication in which offering access to previously downloaded profiles is granted, does not support unpopular profiles to stay available since they are not frequently accessed.

Assuring data availability and integrity is not an issue in F-OSNs, since users may store a single copy of their data only on a reliable professional storage which they trust for not altering or removing their data. Social network architectures relying on flat storage [Famulari and Hecker, 2012] with no OSN-specific logic implemented, allow *flexible choice of storage resources*, since different types of storage resources (e.g. upload and download services, e-mail boxes or personal web space services) exist. Users may choose external servers for data storage based on criteria such as technical specifications (storage size and bandwidth), trust, monetary costs or reliability. The drawback is that there must be a place to process the OSN logic and if it is not the storage offering server, an additional party, which has to be trusted, is necessary.

In contrast, approaches relying on special OSN servers for storing user data [Schwittmann et al., 2013, Seong et al., 2010, Schulz and Strufe, 2013, Anderson et al., 2009] abolish the need for external OSN logic deployment but limit the users in choosing a storage location to the OSN servers instead of any arbitrary storage resource.

Hybrid approaches, allowing both to store on dedicated servers as well as to store on own hardware (e.g. diaspora), relieve users from the need for external services.

4.4.2 Decentralized Access Control

Limiting access to content to a desired set of recipients is at the core of each privacy concept. Three general concepts can be found: ACLs (Access Control Lists), encryption schemes and a combination of both. All those concepts can be realized on the granularity of individuals, role-based access control as well as access control on the basis of a group management system.

Restricting access via ACLs can be realized straightforward via granting access to legitimate users after authentication (before accessing content, the knowledge of the secret needs to be proven) or identities (users being part of a content owner-defined set of legitimate identities are allowed to access).

Since ACLs do not provide any kind of protection against attackers which are able to listen to the communication at the underlaying network and access policy enforcing parties need to be trusted, several authors of DOSNs suggest to implement encryption schemes in spite of their need for key distribution. In our classification, we thus characterize the approaches based on the trusted parties enforcing the ACLs and the adopted key management mechanism in case of content encryption.

Most encryption-based DOSNs [Wilson et al., 2011, Famulari and Hecker, 2012, Baden et al., 2009, Anderson et al., 2009] rely on Out-Of-Band (OOB) mechanisms for exchanging the whole keys (or fingerprints of the key that can be used for retrieving the associated key, for example relying on cryptoIDs [Perrin, 2003], as suggested in [Anderson et al., 2009]). This of course implies the disadvantage of the need for a secure and trustworthy OOB channel.

As an alternative to OOB mechanisms, Safebook [Cutillo et al., 2009a], PeerSoN [Buchegger et al., 2009b] and [Sun et al., 2010] rely on trusted nodes playing the role of *Credential Authorities/PKI*. In those cases, the Credential Authorities (CA) are only used to cryptographically initiate the OSN, while they do not mediate communications and cannot trace interactions.

Finally, Graffi et al. in [Graffi et al., 2008] propose to rely on a DHT substrate also for distributing keys, which allows to avoid any necessity for a central node in the network. As a consequence, there is no need for OOB communication nor Credential Authority, but it comes without any kind of identification.

Cryptographic methods which allow malicious parties to access cipher code still do not prevent from inference attacks. Attackers may infer the type of a message (e.g. video vs. chat) from the size. Furthermore, access to cipher code allows attackers to notice actions (depending on the encryption mechanism) like revocation of access rights or access patterns [Greschbach et al., 2012]. A combination of limiting access via ACLs and encryption or obfuscating methods like chunking and salting can mitigate this issue.

4.4.3 Interaction and Signaling Mechanisms

Different types of interaction handling mechanisms with contrary implications have been proposed by the authors of the approaches which are covered by this survey. Interaction includes messaging as well as sharing pieces of content. Sharing operations consist of making pieces of content available for being downloaded by other users. We distinguish between centralized and decentralized interaction mechanisms.

Centralized Handling of Signaling

Centralized interaction handling can be achieved by purpose-built specialized services or via utilizing existing OSNs (e.g. Facebook) for signaling.

The centralized interaction mediation and handling of metadata is suggested in [Kryczka et al., 2010]. In contrast to centralized OSNs where resources of a central authority takes care of different functionalities (e.g. storage, authentication and interaction management), a (single) central node is responsible just for the interaction and metadata handling. The aim is that interaction handling can be done via reliable and powerful units for achieving good quality of service without having a central authority which is able to access user data. The underlaying assumption is that user data access (like in centralized OSNs) is a crucial part of the service provider's omnipotence and needs to be abolished. Lockr [Tootoonchian et al., 2009], Polaris [Wilson et al., 2011] and Confidant [Liu et al., 2011a] rely on *existing platforms* performing signaling mechanism.

However, centralized interaction and metadata handling approaches still give the OSN provider access to metadata and content access information. This means that the central authority is still able to learn e.g. the interests, social connections and popularity of users and their profiles. Hence these approaches require the users to trust the authority to a certain level.

Decentralized Handling of Interactions

P2P systems often rely on a DHT as a signaling mechanism thus mediating interactions. For example, in Safebook [Cutillo et al., 2009a], PeerSoN [Buchegger et al., 2009b] and Vis-a-Vis [Shakimov et al., 2009] the DHT system can be used as an asynchronous messaging mechanism, which may include signaling of new (inter)actions. Users can query the DHT with the ID of a user or a specific piece of content.

A unique feature of P2P-OSNs is the possibility of direct interactions among users, also with no Internet access, as suggested in PeerSoN [Buchegger et al., 2009b] and in Polaris [Wilson et al., 2011]. While PeerSoN relies on direct data exchange among user devices, in Polaris data may be stored/replicated on external servers and user smart phones only play the role of entry point to user data.

The decentralized interaction handling comes with the main advantages not to require trust into a single entity for that (interaction) purpose. Drawbacks are that decentralized systems may leak metadata by cipher evaluation [Greschbach et al., 2012]. They may also suffer from churn-caused node unavailabilities and - like approaches with purpose-built centralized interaction mechanisms - from missing connectivity to popular centralized OSNs like Facebook.

4.5 DOSN Approaches

This section supplements the Table 4.1 with a short paragraph of text for each approach. The rationale behind this section is that a classification cannot capture all unique details of all approaches. The aspects which are already covered in the classification are not mentioned again, except when they are part of the unique clue of the approach. Advantages and disadvantages are not discussed in this section for each approach, since similarity of approaches causes redundancy in the evaluation. Section 6.2 encloses an evaluation based on classes instead.

Furthermore, we explain the publication time line aiming at making it easy to grasp when an approach was published and which approaches can be assumed to be known by which authors of newer approaches.

4.5.1 P2P-OSNs

PeerSoN

The authors of PeerSon [Buchegger et al., 2009b] propose a two-tier architecture in which the first tier is a Distributed Hash Table (DHT) and the second tier consists of the nodes representing users. The idea is to use the DHT to find the necessary information for users connecting directly to the target nodes. This approach comes without a replication scheme and stores offline messages at the DHT (OpenDHT in the prototype implementation). All user contents are encrypted.

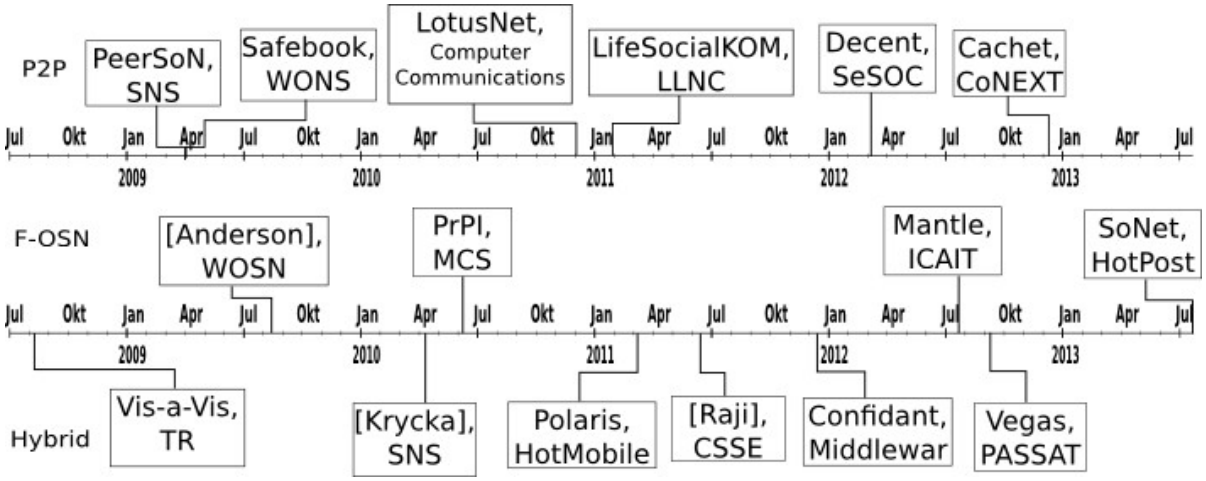


Figure 4.2: Publication date timeline of the surveyed approaches

Safebook

The main objective of Safebook [Cutillo et al., 2009a] is to protect privacy of users in a DOSN setting. The architecture consists of three main components, namely: Matryoshkas (a ring-like ego graph reflecting friendship relations), a P2P lookup service and a Trusted Identity Service (TIS).

Each node is surrounded by its friends (first shell) and friends-of-friends (second) shell in its Matroyschka. User profiles are replicated for better profile availability at friend's nodes in the innermost shell. Nodes at the outermost shell are entry points for routing requests to the center of the Matryoshka and can be found via querying the lookup service. This overlay structure hides the friendship relations from strangers by multihop routing. TIS verifies user identities.

LotusNet

LotusNet [Aiello and Ruffo, 2012] is a modular P2P-OSN platform, realizing social network functionality in widgets. The communication infrastructure as well as the encryption scheme and the identity management is realized by using the DHT modification Likir [Aiello et al., 2008]. Access control is realized by signed grants for proofing social relations. The data owner hence specifies the type of social relation which is necessary to access the data item.

LifeSocial.KOM

Graffi et al. [Graffi et al., 2008] present an approach where the entire OSN functionality is realized by plug-ins. Storing and exchanging data items is realized with the help of FreePastry [Rowstron and Druschel, 2001]. PAST [Druschel and Rowstron, 2001] is used for data replication. Cryptographic public keys are leveraged to be user IDs in the network thus uniquely identify users in addition to encrypt content and messages.

DECENT

Decent [Jahid et al., 2012] is a modular and object-oriented DOSN architecture. It leverages a DHT to store user data and uses cryptography to protect confidentiality and integrity of user-owned content. The focus of the authors is a blog-like wall rather than chat messages. The architecture is modular, i.e. the data objects, cryptography and DHT are separate components interacting with each others based on an interface. This modularity causes freedom to use any kind of cryptography (ABE-based on [Jahid et al., 2011] is suggested in DECENT) and any type of DHT.

Cachet

Cachet [Nilizadeh et al., 2012] is an improvement of DECENT. Thus it is also a decentralized architecture for social networks that provides strong security and privacy. The main difference is that Cachet introduces social caches to improve the performance of the system by avoiding the pull-based grasping of many single data items from different sources. Therefore nodes leverage social trust relationships to "maintain continuous secure (SSL) connections with online contacts to receive updates directly as soon as they are produced". In case of overlapping online times, this type of presence protocol can effectively reduce communication delays.

4.5.2 F-OSNs

SoNet

SoNet [Schwittmann et al., 2013] circumvents the implications of P2P mechanisms (like profile availability and free-riding attempts in resource provision) by suggesting an XMPP-like architecture. Every node is attached to one server, implying the address scheme to be user@host (RFC 822). Profile data is encrypted and replication is still part of the architecture to mitigate server failures. The clue of this approach is to obfuscate the social graph by introducing single-direction pseudonyms.

Mantle

Mantle [Famulari and Hecker, 2012] is a DOSN approach, settled around the idea of leveraging arbitrary storage in the web (cloud services as well as mailboxes, etc.), to store user data. Since the arbitrary storage concept disallows storage entities to deploy any logic, the service-related logic is implemented in user-owned clients. Interaction is managed by employing a publish/subscribe model and is handled locally without any help of a centralized server.

PrPl

PrPl [Seong et al., 2010] stands for Private Public. The main goals are to allow users to store data in their own influence zone by choosing trusted storage resources, and to run social applications across different domains while sharing data without privacy concern. The idea of the architecture is to have "Personal Cloud Butlers" to store personal digital assets to support access control mechanisms. A "Pocket Butler" handles

all authentication and communication with personal cloud butler along with the facility to allow sharing of resources across multiple applications.

PrPl uses SociaLite: a language based on a data log which allows developers to access the data by just querying on the data served by butlers. OpenID is used for authentication.

Diaspora

The main aim of Diaspora[4] is to build a reliable and usable decentralized online social network. The architecture is based (similar to PrPl) on a client-server model where every user has her own server instance (pod) which is used for storage, communication and access control. Since there is no data or service replication, pods must always be online for reliable service provision. A pod can be hosted either on own hardware or by a service provider (cloud service). Data is stored unencrypted on the pod, protected by an access control mechanism.

[Anderson]

The authors of [Anderson et al., 2009] define a privacy preserving architecture for decentralized social networking that takes advantage of the simplicity and performance of the centralized client-server model. The main goals are to protect personal data from unauthorized access, to hide the social graph (like friendship links) as well as assuring content integrity.

The ideas described in this approach are closely related to the field of software engineering rather than network architecture. The authors suggest the client software to consist of the following layers: the application layer, the data structures layer, the cryptographic layer and the network layer. The layered architecture render the software components on each layer exchangeable. All applications are supposed to run in a sandbox, allowing the applications to access just a predefined subset of the private data.

4.5.3 Hybrid DOSNs

Vis-à-Vis

A VIS (virtual individual server) [Shakimov et al., 2009] is a reliable personal server, assigned to every user to store her data. The main idea is to build overlay networks (with VISs as members) that correspond to social groups. Members of groups are supposed to have the intention to share their location. The focus of Vis-à-Vis is to support location-based OSNs while preserving privacy of location information by supporting flexible degrees of location sharing in different groups.

[Kryczka]

In this approach, a User Assisted OSN (uaOSN) [Kryczka et al., 2010] is proposed where users can contribute resources to reduce the costs of the OSN provider and to

[4] https://joindiaspora.com, accessed on 2015-11-01

increase scalability. In uaOSN, queries are sent to the provider that informs the user about storage placement for large content items like photos or videos. The uaOSN provider stores the user profiles and metadata of the outsourced content. Data in uaOSN can be either stored on user's desktop or set top-box/residential router which can have a hard disk or on paid storages like Amazon cloud services. To achieve a better profile availability, data is replicated. An encryption scheme is not part of this approach.

[Raji]

Similar to the uaOSN, Raji et al. propose in [Raji et al., 2011] to store private data (encrypted) beside the OSN on personal storage servers which are assumed to be honest but curious. A BE scheme enforces the access control as well as the confidentiality of the data.

Polaris

Polaris [Wilson et al., 2011] is an "architecture for OSNs that preserves monetary incentives for OSN providers to store and manage user data, while also mitigating the systemic privacy concerns associated with monolithic OSNs." To realize this, a user can choose a different provider for each functionality (e.g. photo storing or micro blogging). Highly sensitive data is stored at a mobile phone which is assumed to be able for keeping small pieces of content available. The authors argue that (as a result) every provider that is involved in service provisioning can just access a subset of the whole personal data.

Confidant

Confidant [Liu et al., 2011a] fosters decentralized data-processing being scalable and affordable by storing data without encryption. It relies on social trust relationships among friends to replicate the data on secure devices. The challenges addressed in this paper are access control and data consistency among the distinct replicas.

Vegas

Vegas [Durr et al., 2012] is a DOSN architecture proposing to use reliable data storages for increasing the availability of user data in a P2P setting. The encryption scheme is based on mutual public keys for exchanging symmetric keys. IT is used to ensure the confidentiality of user data.

4.6 Evaluative Discussion

In this section, we discuss the present situation in the field of DOSNs on the way to become an alternative to their centralized counterparts. We thus elaborate the degree of achievement with respect to our requirements. We focus on the fitness of the DOSN approaches to help to improve privacy as well as on the quality of user experience. We discuss the latter by looking at performance issues, resource provision, technical knowledge which is necessary to use the DOSN and finally the offered functionality.

4.6.1 Privacy and Security

The major reason for authors to suggest a distributed approach for social networking is to increase privacy and security. Hence, the enthralling question is: Are the suggested approaches appropriate to achieve better privacy and security? We discuss this question with respect to our adversary models (Section 4.1.2, Figure 4.3).

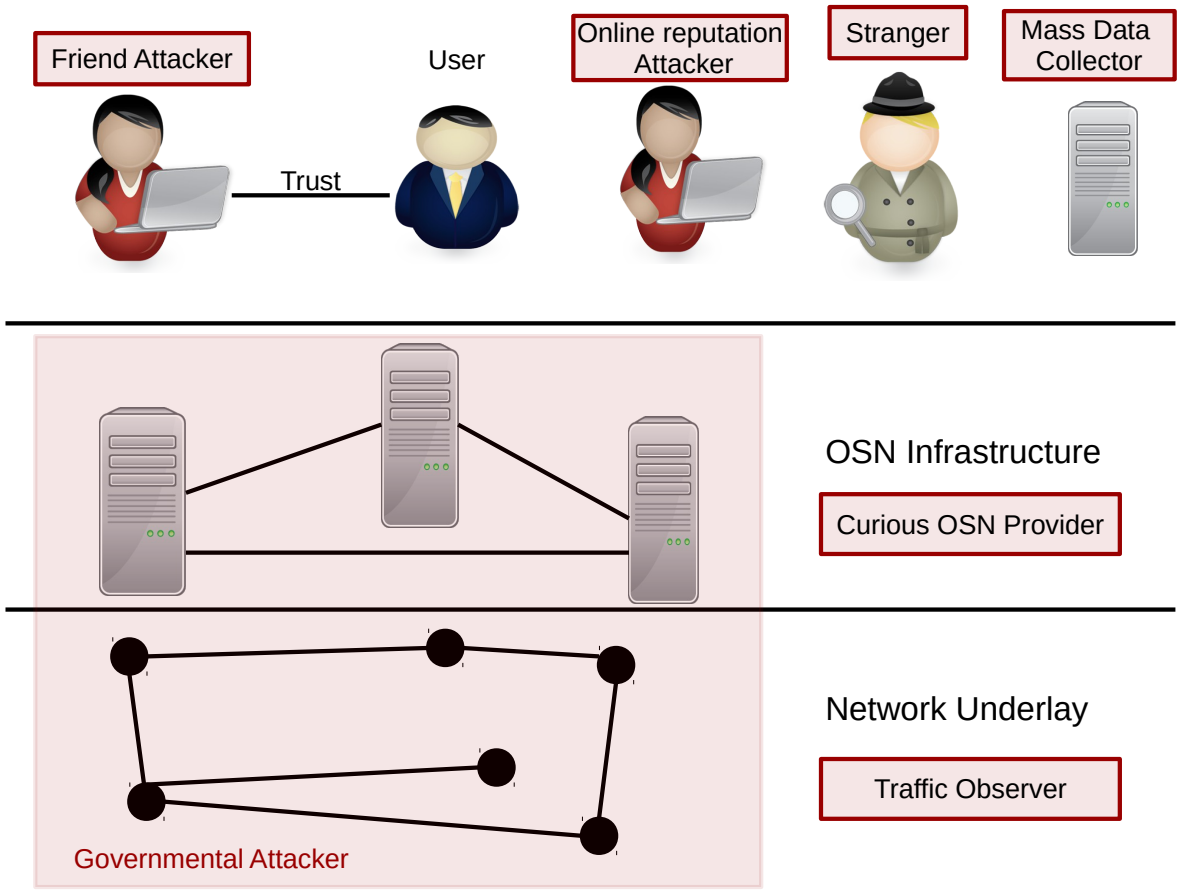


Figure 4.3: Layered OSN model that illustrates the attackers: user, OSN infrastructure, network underlay

SNP Attacker

The overwhelming majority of approaches abolishes the SNP completely and hence it does not exist as attacker anymore.

uaOSN uses user devices for data storage. Even though the users may be able to exactly specify which data is sensitive[5] and store this data on private storages, the OSN provider is still able to learn sensitive facts about the users by evaluating incidental data. The SNP may learn habits like diurnal usage patterns (e.g. conclude that the user works at night) and the social graph. That issue applies for Polaris in the same way, since every provider of a particular functionality can learn usage patterns and two-sided actions like messaging (sender and receiver) potentially leak knowledge about

[5] we doubt that, since sensitivity is depending on the knowledge of the attacker

social relationships. We argue that it is necessary to hide metadata and communication habits as well as social graph information from the SNP to protect user's privacy.

Traffic Observer

P2P-based approaches implement some kind of encryption. Assuming that the attacker is not able to decrypt ciphers, she is still able to infer who communicated with whom and how often. Furthermore, the data item sizes can be inferred by observing the traffic. This allows for guessing what kind of data is exchanged (chat messages, photos or videos). If replication schemes rely on social graph metrics (e.g. friendships or trust), those can potentially be observed as well. Only one approach, covered by this survey (Safebook), tackles these issues by redirection schemes or traffic obfuscation.

Safebook introduces the concept of Matryoshkas where friends form ring-like structures in egocentric networks. Traffic is redirected from outer to inner circles. Nevertheless, Matryoshkas are still vulnerable to timing and traffic observation attacks since there is no traffic obfuscation or message throttling included.

Inferring facts by observing traffic in F-OSN can be challenging if more than one user is using one server and if the servers are re-encoding the data items, since the traffic observing attacker can then only observe that a set of users is connected to the server but not who exactly communicates with whom. The success of the attacker depends on how much information can be learned from communication intensity (traffic, data size) and timing attacks.

Considering hybrid solutions, the situation strongly depends on the concrete architecture. They are as vulnerable to traffic observing attacks as P2P solutions are if direct communication happens among peers or if it can be inferred from the storage place to which the latter is assigned. Approaches like Polaris [Wilson et al., 2011] may mitigate the success of traffic observers by combining different centralized services for communication and storage. uaOSN [Kryczka et al., 2010] cannot be evaluated by now, since it is highly dependent on what exactly is stored at the peers and how it is accessed. A caching mechanism in the centralized part of the uaOSN can be a game changer for traffic observing.

Governmental Attacker

Non-democratic governments tend to try to censor or even disable social networks as soon as riots start in their country. The Arab spring is a prominent example for that phenomenon. Governmental type of attacker can (simplified) be considered being a unification of the SNP attacker as well as the traffic observer. The questions is: Is there an approach which can resist the governmental attacker?

Even though assuming that the government does not want to disable the whole communication infrastructure of a country to disable social networking, we argue that none of the approaches is bullet proof. P2P approaches (without traffic obfuscation) are vulnerable to traffic observing attacks: Governments could find out who communicates with whom and who is important for organizing demonstrations. Server-based architectures can easily be deactivated if the servers are well known and run within the

influence zone of the government. From our point of view, research in making DOSN more resilient against governments is eligible.

Stranger

Stranger attacks with the goal to learn private facts about a particular user are very weak if people would use the access control mechanisms properly and if they would be aware of inference attacks (assuming facts to be valid also for friends). The success of stranger attacks is in general less depending on the architecture of DOSN but rather on efficient access control. Perfect usage assumed, none of the presented DOSNs open an attack surface for stranger attacks.

Mass Data Collector

State of the art for mass data collection is building crawlers. That could be possible if data items are publicly-accessible and if user handles are available to address profiles. The straightforward approach for crawling a social network is to first create a user account then initially connect to arbitrary users and try to crawl their friend lists.

Iterating over friend lists can theoretically lead to a discovery of the whole connected region if every user allows to access a list of her friends. Hence, no matter at which type of architecture, it is very important to disallow strangers accessing friend list in general. DOSN architectures thus do not help to mitigate attacks from mass data collectors in case that access to profiles is restricted in centralized OSNs.

Friend Attackers

Friends, being attackers aiming at accessing more information than authorized by the data owner, can be successful either when access control is not performed properly or if replication schemes in P2P-OSN rely on social trust. A node where friend's (encrypted) content is stored can still learn incidental data.

Cyber Bullying

The social phenomena to attack the reputation of an individual can be observed in an environment like OSN as well. In a centralized setting without content encryption, the omni-potent provider can delete content as well as user accounts if the well-behavior rules are not respected by users. No author of a DOSN considered misbehavior of users by now. To tackle that issue, accountability of actions (e.g. posting content or messages) needs to become a focus of DOSN. Accountability, however, may affect the achievement of anonymity goals.

Conclusion for the privacy and security evaluation: The main advantage in privacy protection, which can be achieved with the approaches in this survey, can be seen in protecting against the central OSN provider. Authors tend to protect communication content rather than hiding communication. From our point of view, no approach can protect against the governmental attacker being interested in building communication graphs.

Censorship is not an issue in most DOSNs (in Polaris and uaOSNs it still is) because of the usage of cryptography. This implies that it is hard to prevent illegal actions. Illegal content can be shared with only a minimal risk and attacks on the reputation of users are abetted. The answer to the general question whether DOSNs improve privacy and security of users strongly depends on the point of view: DOSN can effectively protect the content which is shared but they may foster misusage.

4.6.2 User Experience

Measuring user experience by conducting a user study in DOSNs is hard to perform since no DOSN (except Diaspora) has a user basis rather than public-available and usable prototypes. Having no better alternative, we discuss performance issues, the skills being necessary for using DOSN, and the functionality instead.

Performance

P2P approaches replace the database queries in centralized OSNs by vast inter-node messaging for accessing data. The authors of Safebook [Cutillo et al., 2009a] consider 11 seconds to be a realistic time for requests if no performance optimizations are employed. Cachet introduces a caching strategy for improving the performance by maintaining encrypted channels to friends. Receiving unpredictable data requests from strangers (e.g. friend-of-friend) still causes time consuming operations.

F-OSN and Hybrid approaches do not suffer from these P2P-specific performance limitations, but still cause federation overhead. In general, we would see a performance advantage for centralized OSNs since a single authority is able to globally optimize its databases and to build caches.

Usability

If DOSNs leverage cryptography for privacy, basic knowledge about cryptography may be necessary to use the DOSN. For example, users need to understand that they need to exchange public keys. Furthermore, the presented approaches (except Polaris) require users to install a client software instead of being only a web application. Installing software on local machines may cause a need for actions which require administrative rights on the local system. Moreover, the necessity of local software installations could cause interoperability issues and is an additional procedure which might be an obstacle for users. We thus argue that any kind of DOSN should be running at every web-connected device without installation obstacles to achieve a usability which is comparable with centralized OSNs.

Functionality

No DOSN provides the same functionality like Facebook. One reason of is that the approaches are academic and concentrate on a particular idea to present rather than being intended to be a social network which can be used by the users.

Another reason is more wholesale in nature: popular functionality like recommender systems, search functionality and some online games leverage the social graph and

user attributes. Having only local knowledge, the complete social graph is not known to any single entity. Only local (egocentric) structures can be learned via exchanging messages. Functionality based on global knowledge remains too expensive in any case.

Acquiring the graph knowledge beyond the own graph neighbors (friends) via messaging may effect the user privacy. Even paramount functions like a sophisticated search mechanism for user handles is not available in a privacy preserving manner.

Conclusion of the User Experience Evaluations

Considering our metrics for the user experience, the presented approaches (except uaOSN) will suffer disadvantages in comparison with centralized OSNs. This holds for performance as well as for the required user skills and the functionality. We consider F-OSN and Hybrid DOSNs to have the biggest potential to present a good trade-off: they can be web-based as OSN are nowadays, do not suffer the performance limitations caused by maintaining P2P structures, and in case that the majority of friends is assigned to one server, they allow efficient local operations instead of grasping data items from a high number of different sources.

4.7 Related Approaches

We discuss some approaches in this section which address DOSN related issues or approaches only aiming at improving specific sub-aspects of DOSNs to highlight current challenges. We include approaches addressing the profile availability issue in P2P-DOSNs, encryption schemes for DOSNs as well as for centralized OSN being one alternative to distributing OSN and finally we include social network integrators.

Social network integrators are also included, since they offer potential ways of extending the initially limited user basis of DOSN to temporarily increase their attractiveness until sufficient adoption. We thus think that a social network connector is a crucial success component for new upcoming social networks.

4.7.1 Profile Availability in P2P-based OSNs

The load and requirements for storage in P2P-OSNs differs strongly from distributed storage as well as from file sharing. OSNs environments require the storage layer to reliably store many unpopular content items which are frequently updated. This is in stark contrast to file sharing applications, which usually provide a comparatively low number of large and popular files to a high number of users. A stark contrast exists even to conventional P2P backup and storage scenarios, which are characterized by rather infrequent I/O access to the stored data. Furthermore, contrary to file sharing and P2P backup and storage, in which all participants are treated somewhat equally, OSNs contain information about friendship and trust relationships that can be exploited. Many techniques that are deployed in P2P storage environments - like erasure codes - are not convenient in this dynamic environment. Thus, none of the P2P-OSN approaches is based on a file sharing nor P2P storage scheme. The following

subsection explains the solutions for profile availability in P2P-OSNs that are discussed in the literature.

As we have shown in [Paul et al., 2012a], choosing replication nodes randomly leads to a high number of replicas if a convenient availability of user profile data should be achieved. Friend storage approaches suffer from localization effects: if all friends live in the same time zone (e.g. same city), it is very likely that they have the same off-line times in the night. Furthermore, if a new node has no connections to friends, it does not benefit from replication. Choosing the best subset of friends for profile replication is an NP-hard problem [Sharma et al., 2011].

Finding a systematic solution for having a good availability with minimum cost and overhead is the goal of the authors of MY3 [Narendula et al., 2012], SuperNova [Sharma and Datta, 2012], Gemstone [Tegeler et al., 2011], SOUP [Koll et al., 2013] and S-Data [Shahriar et al., 2013]. These approaches aim to improve the effectiveness and efficiency of replication mechanisms to increase the performance of P2P-DOSNs.

SuperNova introduces super nodes for bootstrapping and circumventing the disadvantages of utilizing friends' nodes for storage. Gemstone has a metric-based approach to select some friends which are a good choice for achieving a high availability with low costs and SOUP proposes to select replica nodes by calculating an online experience among friends. S-Data is a group-based approach where groups are generated on the basis of diurnal online patterns to reduce the number of replicas. The authors of MY3 [Narendula et al., 2012] propose users to choose a subset of friends (trusted proxy set) for performing profile replication and access control. Arguing that trusting the friends which are performing the access control can replace the encryption of content and hence abolishing key distribution. This assumption simplifies the approach.

Common ground of the mentioned P2P-DOSN availability improvement approaches is that the authors assume at least a small subset of nodes to have very long session durations and to be stable in terms of churn. However, the churn behavior that we observed in our user studies (Section 2.2) does not support this assumption. We hence suspect these approaches to provide a lower availability of user profile data in case of assuming the session durations that we have observed.

4.7.2 Encryption Schemes for OSNs

Abolishing the omnipotent and trusted social network provider as a mediator of communication between social network users, combined with the introduction of replication schemes in P2P-OSN, leads to the need for encryption of content and communication (in case of leveraging untrusted resources). F-OSNs or hybrid solutions often aim at mitigating the need for trusting server entities and rely on cryptography for this purpose. Only a minority of DOSN approaches comes without encryption and relies on trust in friends (e.g. My3 [Narendula et al., 2012]).

Thus, the efficiency, performance and usability of encryption and key distribution schemes are crucial factors for DOSNs for being widely adopted. For this reason, some authors work on building new cryptographic mechanisms for that specific issue. Men-

tioning their relevance, we present a brief sketch of the ideas and a brief overview of this field.

Brief OSN Encryption Background

Using mutual public keys is the straight forward way of realizing an encryption scheme. For every recipient of a message, one encryption procedure has to be performed. Asymmetric cryptography, however, is comparatively expensive compared with symmetric cryptographic algorithms. Group key management mechanisms based on symmetric keys tackle that issue by distributing one symmetric key among a group of recipients (e.g. via mutual public key schemes). Hence, one (symmetric) encryption process is sufficient to share content with a group of recipients. As long as the group setting does not change, a new key distribution is not necessary.

Broadcast encryption (BE) schemes can rely either on symmetric or asymmetric encryption and are used by senders to share confidential data with a dynamic set of recipients in a cost-effective way. BE requires each recipient to have an individual key. In BE schemes, the encryptor uses an encryption mechanism that allows to produce ciphertext that can be decrypted by plenty of keys which are defined during the encryption process. If a private key generator, which is leveraging identities to decide about legitimation, is part the system, the broadcast encryption scheme is called identity-based broadcast encryption (IBBE) [Delerablée, 2007, Boneh et al., 2005].

Attribute-based encryption (ABE) [Sahai and Waters, 2005, Goyal et al., 2006] schemes adopt that idea. An encryptor decides who is able to decrypt the ciphertext by labeling the latter with a set of descriptive attributes. Private keys are associated with ACL structures to decide, based on those attributes, which ciphertexts can be decrypted. The encryptor thus does not decide about decryption by taking single keys or identities into account but defines attributes or combinations of attributes which a decryptor needs to meet to be able to decrypt a message.

(D)OSN Encryption Contributions

The main goal of Persona [Baden et al., 2009] is to disallow third parties from accessing personal information by deploying attribute-based encryption (ABE) in an OSN context. Each user generates an ABE public key (APK) and an ABE master secret key (AMSK). For each friend, the user can generate an ABE secret key (ASK) corresponding to the set of attributes that defines the groups that friend should be part of.

The main contribution of the model from Sun et al. [Sun et al., 2010] compared with Persona is to have a very efficient revocation of content access rights. It uses broadcast encryption that enables the data owner to exercise desired access control.

The authors of Noyb [Guha et al., 2008] (“none of your business“) suggest to improve privacy by encrypting content and to modify it in a way that it looks like legitimate content. Hence, it allows users to use existing OSN while disallowing provider to access the content. Applying this approach is not an obstacle for the provider to learn usage patterns as well as friendship relationships.

Günther et. al. [Günther et al., 2012] provide a building block for privacy preserving treatment (including encryption) of user profiles in OSNs. The authors of [Bodriagov and Buchegger, 2012] compared different encryption mechanisms and evaluated them for their applicability in the DOSN context and concluded that broadcast encryption would be the best choice for this use-case.

EASIER[Jahid et al., 2011] is an ABE architecture for DOSN, supporting dynamic group memberships and revocation of rights without re-encrypting data items or issuing new keys. The main idea is to introduce a proxy which needs to be contacted before decryption. A user sends a part of the cipher text (CT) to the proxy where a transformation takes place. The transformed CT can only be decrypted if the right was not revoked.

Lockr [Tootoonchian et al., 2009] is an identity-management tool for OSNs that allows users to codify their relationships through social attestations. The primary goal is to provide privacy as well as to simplify site management and accelerating content delivery. Lockr's decoupling eliminates the burden on users of maintaining several up-to-date copies of social networks, performing user-ID reconciliation across sites, and familiarizing themselves with the varied access control mechanisms provided by each site. A social attestation is a piece of data that certifies a social relationship. By issuing an attestation, the issuer tells a recipient that they have formed a relationship.

The presented encryption approaches show that retaining confidentiality of content in OSNs is possible. In case of suggesting a new DOSN approach, authors may rely on existing encryption mechanisms.

4.7.3 Private Discovery of Common Social Contacts

Discovering common social contacts is a common feature in today's OSNs like Facebook. In privacy preserving distributed systems, it may not be desired to exchange contact lists. De Cristofaro et al. [De Cristofaro et al., 2013] introduces a scheme which allows for finding common friends without disclosing non-common friends. To the best of our knowledge, there is no discovery mechanism which neither disclose the search index nor the search queries. As a result, there is a trade-off between implementing (or using) a discovery mechanism or preserving privacy with respect to search index and queries.

4.7.4 Social Network Integrators

OneSocialweb[6] is a project aiming at building an XMPP-based connector which potentially integrates all OSNs into one large social network. Other social network integrators, connecting a subset of popular services are:

1. Meople (http://meople.net/, accessed on 2014-01-20) aggregated SNSs: Facebook, LinkedIn, Google+, Twitter, Instagram, YouTube, Flickr, Groupon, Tumblr, Foursquare, VK, Odnoklassniki

[6] http://onesocialweb.org/, accessed on 2014-01-20

2. Jyst (http://jyst.us/, accessed on 2014-01-20) aggregated SNSs: Facebook, Twitter
3. Alternion (http://www.alternion.com/, accessed on 2014-01-20) aggregated SNSs: Facebook, LinkedIn, Twitter, Instagram, YouTube, Flickr, Foursquare, Picasa and the mail accounts: Gmail, Hotmail, Yahoo!, AOL
4. Yoono (http://www.yoono.com/, accessed on 2014-01-20) aggregated SNSs: Facebook, LinkedIn, Twitter, YouTube, Flickr, Foursquare, MySpace, Yahoo!, Google Talk, AIM, FriendFeed
5. TweetDeck (http://www.tweetdeck.com/, accessed on 2014-01-20) aggregated SNSs: Twitter, Facebook, Myspace, Linkedln, FourSquare, GoogleBuzz
6. Hootsuite (http://hootsuite.com/, accessed on 2014-01-20) aggregated SNSs: Facebook, Linkedln, Foursquare, MySpace, PingFm, Wordpress
7. SpredFast (http://spredfast.com/, accessed on 2014-01-20) aggregated SNSs: Facebook, Twitter, LinkedIn, Google+, YouTube

The aforementioned social network connectors could potentially been leveraged to bootstrap a new (D)OSN since the attractiveness of OSNs is strongly bound to the user basis.

4.8 Decentralization Impact on Stakeholders of OSNs

In this section, we discuss the impact of decentralizing OSNs on today's OSN stakeholders. This includes benefits, drawbacks and challenges.

The stakeholders in the context of OSN, considered in this work, are: the OSN users, the OSN provider, the advertising companies which benefit from utilizing the advertising opportunities offered by the OSN provider, the governmental state and media consumers which are not necessarily part of an OSN. Effects on extenders (e.g. application sellers) like Zynga[7] are not discussed since the effects on them is strongly depending on the particular architecture.

Since we consider the user to be the most important affiliate, we start our discussion with the following benefits of OSN decentralization for the users:

- *Ownership:* Facebook and other OSN providers ask the users to transfer the copyrights of any content from the user to the OSN owner. In contrast, decentralization holds user data in the influence zone of the users. The copyright transfer can be avoided.
- *Privacy:* In centralized OSNs, users need to trust the omnipotent provider not to misuse the data and to be able to protect the data from attackers.
- *Flexible choice of resources:* Building, running and maintaining OSN platforms cause expenditures. In centralized OSNs, platform-related resources are provided by the service provider itself. In most cases, they present no monetary costs to the final users [Falch et al., 2009], which pay by agreeing for such platforms to exploit their data with a commercial purpose.

[7] `http://zynga.com/`, accessed on 2014-07-06

One of the benefits a decentralized approach should bring to users, is that they should have the possibility to choose what resources to rely on. For example, a user can choose whether adding the own device's resources (e.g. P2P approaches) or relying on dedicated servers (e.g. Diaspora). Using dedicated resources for building a DOSN does not cause an exploitable dependency like using the resources of centralized OSN provider, since the OSN membership does not require an affiliation with one specific authority. Hence, the resource provider is replaceable. This opens to several business models which strongly differ from exploiting user data for commercial purposes.

- *No censorship:* Several centralized OSNs perform active censorship[8,9] - called decency or content control - on user behavior and content. This imposes significant limitations on what a user can or cannot do within the platform, based on rules which may strongly vary from one platform to another and are in general very subjective. Such rules can also be very country-specific, since OSNs have already accepted to be compliant to local laws imposed by several governments.
- *Openness:* DOSNs abolish the central authority, causing that no single authority is able to exclude a user from the platform by suspending his account (e.g. because of not accepting copyright transfer rules).

While decentralizing OSNs does not come without drawbacks, the following aspects may become an issue for users:

- *Resource provision:* Centralized as well as decentralized OSNs need technical resources (e.g. storage, bandwidth) to operate. In centralized OSNs, the provider is responsible for making them available. The most popular approach to compensate these efforts is to sell opportunities for personalized advertising. In the decentralized case, other mechanisms need to tackle the resource issue.
- *Profile availability:* Availability of user profiles is strongly linked to the OSN architecture. In the centralized case, the provider takes care of storing and keeping user profiles available. Federated DOSNs rely on independent professional and reliable resources while P2P OSNs utilize the unreliable resources of the users devices.
- *Cyber bullying:* Since decentralized OSNs hide or encrypt the communication, no central authority can enforce rules. In the real world, no administrator can switch off the voice of people which are bullying others as well, but everybody is responsible for what he or she is doing. We argue that DOSNs should support accountability to protect users.
- *Metadata privacy and the concealment of communication partners:* Depending on the particular architecture, decentralizing OSNs may raise security issues that do not exist in centralized OSNs [Greschbach et al., 2012]. An attacker which is able to observe traffic in the underlying communication network (e.g. IP) can track who communicates with whom if OSN devices are assigned to a single user and can send messages to other devices without obfuscation. In contrast, a centralized

[8] `http://www.facebookcensorship.com/`, accessed on 2014-01-20

[9] `http://www.dailytech.com/Google+Plays+Name+Police+Conducts+Baffling+Censorship+Crusade/article22238.htm`, accessed on 2014-01-20

OSN has a mixing functionality. Assuming frequent usage, the mixing functionality implies that the observer could only find out that users communicate with the provider but not track single communication paths among users.

- *Functionality:* OSNs allow to build social applications which are based on interactions among users who share social links or special interests. Examples for these applications are recommender systems, interest matching algorithms for mediating between users as well as games. Due to the nature of decentralized systems, the complete social graph as well as the complete set of interests of all users is unknown to anybody. Each node has only local knowledge. Hence, the graph knowledge needs to be grasped using federation protocols, if it is necessary for an application.

Other affiliates become more affected by economical issues. Abolishing the central OSN provider naturally destroys their business model and hence other companies cannot benefit anymore from utilizing the advertising opportunities.

4.9 Summary and Conclusion

In this chapter, we discussed the impact of decentralizing OSNs on different OSN affiliates of today's popular OSNs, explained the design space by introducing an architecture model, presented and discussed a classification of DOSN approaches and introduced some DOSN-related approaches. Finally, we presented the unique ideas of each discussed approach.

Decentralization of OSNs can tackle two important issues: First, it is a possibility to circumvent the need to trust the SNP for not learning facts which cannot be hidden by encryption. An omnipotent provider could still learn who communicated with whom and how often. Second, users do not need to accept copyright transfers to the SNP and terms of usage which are disadvantageous for them.

The result from security perspective is that DOSNs mainly aim at protecting content from curious provider and assuring confidentiality of user communication. DOSNs potentially abolish content censorship by leveraging encryption schemes. Beside Safebook, none of the presented approaches introduces mechanisms to protect against traffic observer or governmental attackers building communication graphs.

The result of our functionality discussion is that the discussed ideas do not solve the issue of providing attractive social-graph based functionality like a comprehensive privacy preserving search and a recommender system like in the centralized OSN counterparts. Hence, we argue that the field of DOSNs could benefit from research in building privacy preserving graph-based functionality, combined with performance optimizations.

Chapter 5

Finding User Handles with Privacy

Decentralization removes any single entity with complete knowledge about OSN users as well as about the social graph connecting them. However, several appealing functions in today's OSN, like Facebook, require knowledge about user data and the social graph. These functions include not only advanced inference or recommendation features, but even paramount functions like discovering other users. Even the most popular DOSN, Diaspora, consequently relies on out-of-band communication for exchanging user handles, which are required to connect to other users [Schulz and Strufe, 2013]. This lack of competitive features in DOSNs has severely hindered their success. Existing technologies do not seem to offer appropriate solutions to the privacy preserving user discovery problem. In particular:

1. Creating a central index requires a trusted party, or leak information about the participants. Beyond privacy concerns, economical reasons prevent its setup in a decentralized system, as it would cause costs that are unlikely to be paid for by the users, in the current Internet ecosystem.
2. Classical P2P search assumes a public, distributed search index, which at least partially is stored on and forwarded by presumably untrusted nodes [Raiciu et al., 2009], and commonly even spreads information by replicating its content.
3. Public key encryption with keyword search (PEKS) [Boneh et al., 2004] requires possession of the cipher, and hence a centralized, encrypted index, which is unviable as mentioned above. Applying PEKS to a DHT does not seem viable since the cipher isn't known to the requesting party and hence can not easily be addressed and discovered.
4. Secret database querying approaches do not solve the problem of avoiding to build a database that contains user-linkable information.

In this work, we propose a mechanism that adds the functionality of user handle discovery in RFC 822[1] based systems, to make a step towards providing usable DOSNs with a competitive feature set, while maintaining the privacy of their users. This implies that no information that can be linked to any participating individual may be

[1] https://www.ietf.org/rfc/rfc0822.txt, accessed on 2014-02-18

leaked by the scheme. Because of potential abuse of profile discovery, the scheme must also prevent mass collection of profiles and their identifiers to avoid SPAM or other types of illegitimate messaging.

To tackle this issue, we propose a solution that divides the service into three separate parts:

- The collection of distributed information about servers that host profiles which potentially include a specific user, identified by separate user attributes.
- A privacy preserving negotiation protocol to prove knowledge about the sought user.
- The provision of ephemeral handles with limited validity, which allow for a single message within limited time.

The scheme is applicable to arbitrary hybrid DOSN architectures. These consist of decentralized servers that are equal by design and operated by diverse parties, and are selected by participants to host their profiles. Hybrid DOSN architectures circumvent the implications of leveraging unreliable resources (P2P) while still operating without a central authority. Examples for this type of approaches are Diaspora, Vis-a-Vis [Shakimov et al., 2011], Vegas [Durr et al., 2012] and SoNet [Schwittmann et al., 2013]. Without global knowledge about other servers, a common choice of identifying users and their profiles is to use addresses of the form [user]@[host], following RFC 822. The host part uniquely identifies the server (usually using its DNS name), and the user part the respective participant registered on the server.

The first step towards building our search scheme is to map all atomic user properties on their registering servers. A DHT spanning all participating servers then is used to register the user properties under their server address. Hence, there is no link between the attributes that describe an individual and the user herself, but only a link to her server. Her server consequently can be discovered, when searching for the user, and contacted for further negotiation.

The contacted server then can verify the validity of the discovery request. Demonstrated with knowledge about the target subject (in a privacy preserving manner), the server can decide whether to create a valid temporary handle (Search_ID) for contacting the subject once, or to create an invalid one, pointing nowhere. The participants hence have the liberty to define a selected set (or subset) of knowledge that is needed to discover a valid temporary handle for their profile.

The contribution, presented in this chapter, is to adapt well-known techniques like DHTs, indirection schemes and secret sharing in an innovative and beneficial way to allow finding user handles in decentralized communication systems without facilitating SPAM. In contrast to previous solutions that leverage lookup services, we avoid to build a search index that can be maliciously exploited.

The rest of the chapter defines the requirements and the protocol in Section 5.1, explains our system design in Section 5.2, gives a detailed evaluation in Section 5.4 and finally summarizes the main contributions in Section 5.6.

5.1 Requirements

The central objective is to enable users in a distributed client-server environment to find user handles of communication partners without knowledge about the partner's responsible server. State of the art services are not widely adopted (e.g. public e-mail address catalogs) because of their potential facilitation of copious undesired messaging. Our scheme hence needs to meet the following requirements:

1. The service must not jeopardize the privacy of any participant, and hence no linkable data may be published.
2. The discovery scheme has to be scalable to handle large numbers of users (comparable with popular OSN) and servers.
3. The search protocol must be resistant to illegitimate user discovery. Permanent addresses, or user handles, may be retrieved through the system only upon explicit approval by the related individual.
4. The search protocol must be resistant to unsolicited mass communication. It specifically has to prevent uninformed mass address retrieval.
5. Supporting requirements 3 and 4, the scheme shall merely provide ephemeral alias addresses for single use only.

5.2 System Overview

This section describes the system design and illustrates the solution space. We explain the mechanism to discover the server that is responsible for a targeted identifier, while meeting requirements 1 and 2 and subsequently specifying the registration process of profile attributes. To address requirement 3, we introduce an access control mechanism. Requirement 4 is fulfilled by restricting the served identifiers to be valid within a short term (minute scale). Requirement 5 is met by introducing an ephemeral user handle.

5.2.1 Discovery Mechanism

As a consequence of requirement 1 as well as the absence of a central authority in our system environment, we can neither build a central index to locate user handles, nor use a distributed lookup service which publishes or replicates data to discover them. Since the servers in our scenario are equal in functionality, a potential search request sender does not have a specific location to start the search procedure. Furthermore, privacy preserving negotiation for demonstrating knowledge about the search target to distinguish between legitimate and illegitimate requests is an expensive procedure (with respect to communication overhead).

Thus, the basic idea is to first locate servers which potentially host the sought user handle and subsequently perform the negotiation with a small set of servers. We use an efficient discovery mechanism which allows us to register tuples of attributes and server addresses for that purpose. Promising candidates are DHTs (e.g. CHORD

[Stoica et al., 2001], CAN [Ratnasamy et al., 2001]), since they are scalable and churn resistant.

Our approach implies the need to *register* a tuple of every field (just once per attribute value no matter how often it is occurring on the server) of the user profile (e.g. name, city) and the server address at the lookup service by its own.

Thus, a request to the DHT returns a list of servers, matching the field content. The servers which are part of every single list are candidates which may be the server hosting the desired contact. We will call this list the *candidate list* for the rest of the chapter. This list can then be condensed by intersecting candidate lists for several search requests while increasing the detail of the search. But it terminally may yield more than a single server.

To mitigate the issue of index poisoning, the tuple of server address and the piece of search content can be signed by the server. Registering this signature together with the tuple allows us to validate whether the registration of an item is originated by the legitimate server or not.

5.2.2 Access Control

Each user defines a set or range of knowledge that is necessary to discover herself. A requesting individual then has to demonstrate at least this minimum knowledge about the sought subject in order to get a valid Search_ID. We define a privacy preserving negotiation protocol that allows the search request sender to prove knowledge about the search target without disclosing the query to anybody but the servers, sharing the same knowledge. This prevents the untrusted nodes in the lookup service from being able to learn valid attribute combinations from search requests.

5.2.3 Ephemeral User Handles

Each provided valid Search_ID can only be used for a single message and just for a short period of time. We decided to not provide the long term valid addresses (IDs) of the subject to fulfill the misuse and access control requirements. Being contacted via short term identifier, the sought user handle owner may decide whether to reply or not on the message for disclosing the permanent user handle.

5.3 Protocol

This section gives a detailed description of our protocol, which consists of the two parts of user registration and discovery. It relies on a lookup service (DHT), which is adopted by the communication servers (Diaspora, SMTP, XMPP, etc.) of all participating users.

5.3.1 Definitions

This subsection defines terms for later use in the formal protocol description: *Servers* are nodes, participating in the DHT as well as providing the underlying communica-

tion service (eg. e-mail). *Search Fields* are tuples of properties comprising of a field name and its content, which describe a detectable individual (field "First Name" containing "Bob", for instance). Search fields are filled by participants in order to describe themselves. *The Client* is a part of the software installation. It is installed on the user's machine and is responsible for communicating with the assigned server. *The Ephemeral User Handle* is a string, which is a valid address for just one message in a short predefined period. It consists of a random string which is not guessable.

5.3.2 User Registration

An individual user $\bar{A}$, registered at the responsible server A, fills m of n search fields $f_1, f_2 ... f_m$, describing its profile with strings $a_1, a_2 ... a_m$. $\bar{A}$'s client then sends the data to the server A, which in turn registers itself for each of the descriptors (cmp. Algorithm 1).

Algorithm 1: Registering Search Fields

Data: m of n search fields $f_1, f_2 ... f_m$ of a user profile with strings $a_1, a_2 ... a_m$ as their values
Result: m in the lookup service registered search fields

foreach (f_i, a_i) **do**
  $\tilde{a}_i = concatenation(f_i, a_i)$;
  $h_1 = hash(\tilde{a}_i)$;
  $register\ h_1\ at\ the\ DHT$
end

5.3.3 User Discovery

A requesting user $\bar{A}$ describes the target subject by providing as many specified search fields $\left(\{(a_1, f_1), (a_2, f_2), \ldots, (a_j, f_j)\}\right)$ as possible. $\bar{A}$'s client subsequently submits the entered information to its responsible server A, which in turn retrieves a list of candidate servers, which are responsible for users matching any of the specified Search Fields (cmp. Algorithm 2).

Having a list of servers with potential matches (the length is depending on the popularity and distribution of the search items), the user can submit requests for receiving short term IDs to servers from the candidate list. These requests contain all available knowledge about the target individual. Servers receiving this request check if a matching individual is registered and if this individual's access control policy is met by the request, i.o.w. whether the presented knowledge is sufficient to generate a Search_ID.

The search protocol is depicted in Fig. 5.1, with the following variables:

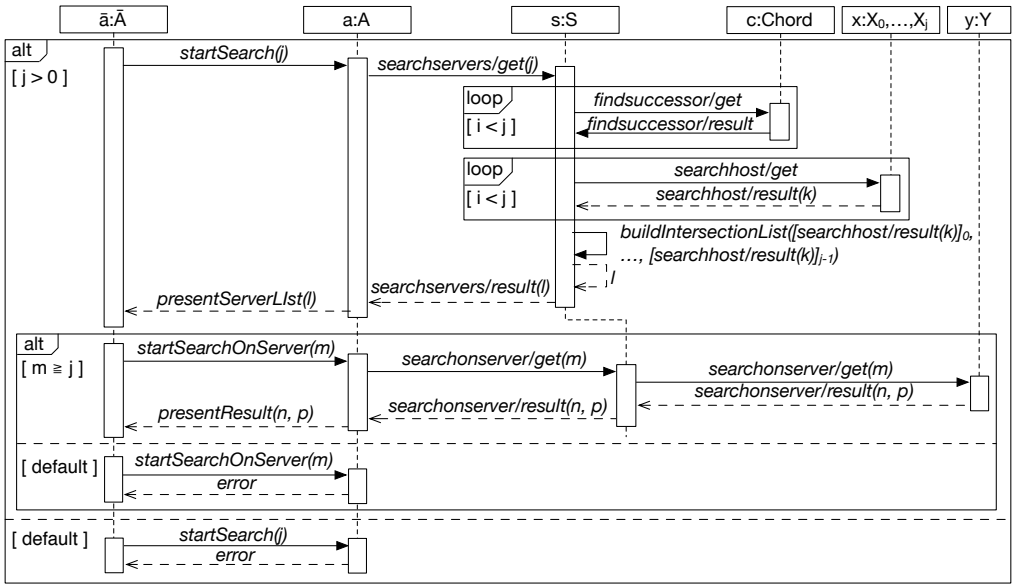


Figure 5.1: Sequence diagram of the message flow during the search phase. $\overline{A}$ represents the user's Client and S the user-assigned server. $CHORD$ illustrates the lookup service. $X_n ... X_{n-1}$ are the servers, selected during the *findSuccessor* operation. j, k, l, m, n and p are the numbers of the variable parts of each message.

j	number of filled out search fields in the search form at the first step
k	number of hosts having at least one user matching one search field
l	number of hosts having at least one user matching all search fields
m	number of filled out search fields in the search form for a specific host (second step)
n	number of matching users on the specific server
p	sum of all public profile fields

Algorithm 2: Search algorithm, conducted by the searcher's hosting server

Data: search fields $f_1, f_2 ... f_j$ with contents $a_1, a_2 ... a_j$
Result: List of servers which potentially host the target subject

foreach (f_i, a_i) **do**
  $\tilde{a}_i = concatenation(f_i, a_i)$;
  $h_1 = hash(\tilde{a}_i)$;
  $request\ h_1\ at\ the\ DHT$;
  $receive\ list\ a_i^l\ of\ servers,\ assigned\ to\ a_i$;
end
$candidate\_List = \bigcap_{i=1}^{n} a_i^l$;

5.3.4 Privacy Preserving Negotiation Algorithm

In case an adversary runs a server and registers popular data items (e.g. popular names) for advertising herself to become part of the candidate list of a user's request, she could misuse the negotiation process for learning more attributes about users. We counter this potential data leakage by suggesting the following negotiation protocol, consisting of two algorithms: one on the server side and one on the searcher's client side.

In this protocol, we propose to define a set of obligatory base attributes: name, first name and city. Our experiments (Section 6.2) show that this combination is a good choice for a real world user identifier. Furthermore, the entropy of a single attribute is small and an attacker could guess the existence of combinations of popular attributes. Our attribute concatenation (Algorithm 3) increases the number of possible combinations per server to the total number of first names multiplied by the number of last names and the number of cities. Hashing the concatenated strings keeps them secret but still allows conducting pattern matching operations on the server.

Please note that if a user realizes attackers guessing the combinations due to the fact of a popular name and a big city etc., she can easily define the minimum knowledge to contain more attributes (e.g. interests, employer) and thus make guessing harder.

Algorithm 3: Client-side negotiation algorithm

Data: m of n search fields $f_1, f_2 ... f_m$ of a user profile with strings $a_1, a_2 ... a_m$ as their values and the Server_ID of the request recipient
Result: Request message content for a temporal user handle

$base = concatenation(a_1 ... a_3, Server\_ID)$;
$P(a_i) = powerset(a_i), i > 3$;
foreach $(x \ in \ P(a_i))$ **do**
    $conc_x = concatenation(base, x)$;
    $knowledgeproof_x = hash(conc_x)$;
    $add \ knowledgeproof_x \ to \ the \ knowledgeprooflist$;
end
$add \ hash(base) \ to \ the \ knowledgeprooflist$;
$Request \ message = knowledgeprooflist$;

We expect the server to have a table of user handles and the corresponding hashes, related to those users which are registered at this location. The hashes are computed according to the user defined minimum knowledge. Subsequently, the server algorithm runs a local lookup for the hashes in the "knowledgeprooflist" (the hashed representation of the requester's knowledge and a result of Algorithm 3) from the negotiation request. If a match happens, the request is replied with a valid temporal user handle, otherwise with an invalid one.

5.4 Evaluation

With respect to our requirements, we evaluate the functionality, the mentioned privacy properties and the scalability of our search protocol in this section. Regarding functionality, we answer the following questions: Is it possible to find persons with three to five attributes (e.g. name, first name, city, employer) in the real world? Is the approach feasible to achieve this? Our privacy discussion answers those questions i.e. whether the protocol leaks private data. Regarding scalability we discuss: How many servers need to be contacted for negotiating with them? How much data is exchanged by applying the protocol?

5.4.1 Functionality

The only real world data collections we are aware of and which are suitable for estimating the search success of a public user handle search algorithm are phone number search engines[2,3], social networks and web search engines (like Google). Thus, we used them for estimating the amount of data which is necessary to identify a person.

Our experiments with the authors of this chapter's conference publication as an example, showed that the first name and the last name as search strings reduce the result list to the length of two (Thorsten Strufe, supervisor) and seven (Marius Hornung, student). Thomas Paul is a common name, leading to the necessity of taking the city information into account.

Further real world experiments in existing user handle lookup services (see above) with about two dozens of popular German names showed that the combination of first name, last name and city usually is sufficient to find the subject. We argue that this sample size is big enough, since drawing two dozen strangers is unlikely and a search service is still useful even if a negligible minority of subjects could not be uniquely identified, just using commonly available knowledge (e.g. city, name, employer). The answer to the evaluation question whether our profile description is applicable hence is true.

To show the feasibility of our scheme, we introduce an example scenario: We assume the server landscape to have a similar structure like the e-mail system or the XMPP system because of both a lack of reliable user and usage quantifications from DOSN and the similarity of the architectures. That means for our scenario to assume some big hubs with plenty of users, providing publicly-available services (e.g. Google) as well as a significant number of smaller servers[4,5] maintained by companies or non-profit organizations like universities. Algorithm 2 builds on the assumption that not every server has registered every search string (e.g. name) which occurs in the system. We assume that the server candidate list usually contains the big hubs, but just a small fraction of small servers.

[2] http://www.dastelefonbuch.de/, accessed on 2014-02-18
[3] http://www.teleauskunft.de/, accessed on 2014-02-18
[4] http://www.mailradar.com/mailstat/, accessed on 2014-02-18
[5] http://www.zdnet.com/jabber-numbers-overtake-icq-3039117160/, accessed on 2014-02-18

We make the following assumptions for our scenario:

- The lookup service (e.g. DHT) works well in low-churn environments like ours, since this concept is well known, evaluated in previous work and tested within our prototype in a small scale.
- The server popularity has the following properties which reflect the situation in the e-mail provider landscape:
  - 10 big hubs (like Google or Microsoft in the e-mail environment) concentrate the majority of users on them,
  - 10,000 smaller (about 100 users in average) servers exist, maintained by companies, organizations and communities,
  - 10,000 different values of each: the first names, last names and cities are known in the system.

The worst case assumption, is to assume that every first name, last name and city name occurs in each of the big hubs. Subsequently, applying our algorithms in our scenario would mean to have a candidate list, consisting of less than 110 servers (out of 10,010) before starting the negotiating process.

The affordability of the negotiation process is shown in Section 5.4.3. Contacting those (less than) 110 servers results in a list of ephemeral user handles where the overwhelming majority is pointing nowhere. Sending a contact request with a short reasoning for the contact request may motivate the desired search target to answer the request and hence disclose the permanent user handle. Please note, as shown before, the combination of first name, last name and city describes a person very well. Thus it is not likely to receive a plenty of contact requests, addressed to the wrong subject. Nevertheless, in case of receiving too many requests, caused by e.g. a popular name in a big city, the burden of proving knowledge can be raised by adding more fields to the knowledge proof.

The presented search algorithm mitigates the problem that the peers, responsible for the most popular keys, become a bottleneck (Zipf distribution of the request popularity [Gummadi et al., 2003], [Breslau et al., 1999]), by saving only the server's address once per popular attribute instance (e.g. popular name on one server) instead of saving each instance of the user attribute separately. Furthermore, our approach circumvents the need to store replicas of fully qualified user descriptions at other peers in the network, as proposed in [Cutillo et al., 2009c] or [Rzadca et al., 2010].

5.4.2 Privacy and Security

This section contains privacy and security contemplations of the proposed protocol as well as further improvements which address security concerns beyond the scope of the system design to mitigate service availability attacks. The scope of this part of the evaluation is to answer the question whether the use of the search scheme can be harmful for the user or server authority. Leaking data could affect user's privacy, coming with negative side effects far beyond the search functionality.

Attacking additional functionality just brings back the actual status before the search protocol was available. Hence, we consider attacks on the search functionality not to be an obstacle for adding this search protocol to a communication system. We thus assume an attacker that aims to leverage the search scheme to illegitimately access private information. Since we trust friends not to publish user handles, arbitrary nodes in the network, except the own server or trusted friends, can be attacker.

The proposed search protocol does not include actions, sending any personal related information to anybody but the server, the user is assigned to. Other nodes, like the nodes in the DHT, just learn that a user named "Bob" is assigned to server *A*. The link between the public string "Bob" and the ID of the subject, which was sought after, has to be made by the server *A*. Hence we meet the requirement 1.

Even blind testing of attribute combinations does not give reliable information about single users, since knowing that both a user with a first name "Bob" and a user with a last name "Brown" exist, does not yield concluding that "Bob Brown" exists. The first name item and the last name item may belong to different subjects, using the same server. Armknecht et al. [Armknecht et al., 2014] show that the entropy of the user attributes is high enough to make guessing (dictionary attacks) unfeasible as soon as a combination of attributes is used as a key.

Requirements 3 and 4 are met by introducing the short term valid ID, since a requesting node needs to negotiate for receiving this ID, which is invalidated after a short time.

Since the design of our protocol is based on the assumption that attackers have less knowledge about the search target than legitimate request senders, this mechanism is not able to prevent well informed attackers from getting a short term valid ID. We argue that this is not an issue, since popular OSNs like Facebook allow strangers to send friend requests as well. To mitigate that issue, we suggest to allow just one short term valid ID per requester ID. Please note that the well informed attacker still does not have access to the permanent user handle (e.g. email address) until she gets a reply message.

Two types of index poisoning are possible. An attacker could register plenty of entries for another server to increase its load caused by being involved in unsuccessful search requests. Furthermore, an attacker could create fake (sybil) IDs to register plenty of fake targets aiming at hiding the real search target. The first poisoning attack can easily be tackled by signing the DHT entries, the latter can be mitigated by reputation schemes and tackled by abolishing the zero cost environment.

Further security limitations are the following:

- Attackers can learn users not to be part of the system if one of her attributes is not popular by sending a request to the DHT. If the server list is empty, the search target is most probably not registered.
- The number of participating servers can be estimated by requesting popular attributes and calculating the super set of all server list items.
- The popularity of servers can be estimated by evaluating the relative frequency of servers being part of the candidate list resulting from arbitrary requests to the DHT.

To summarize: Users can register to the search infrastructure without fearing to publish their user handles or any other private information. However, the functionality of the search scheme can be attacked like in other distributed search schemes as well. We argue that the presented limitations are not an obstacle for this scheme to be adopted in DOSN or XMPP systems.

5.4.3 Communication and Computational Costs

Compared with centralized approaches for finding user handles, decentralized solutions require more communication. One reason is that the searcher does not know both the user handle itself as well as the particular server to query. Hence, the distributed search procedure contains an additional step for finding the responsible server. Despite this overhead, any useful search scheme must not become prohibitively expensive in terms of network and system load (requirement 2). In the latter of this section, we show that the network load is not a prohibitive issue. The formulas can be found in the appendix.

Having a given search field distribution (e.g. names and city inhabitants) and user-server affiliation frequency distribution, the final network load per search request thus depends on the efficiency of the chosen lookup service, the complexity of the search request, the number of participating servers and the number of system users (and hence are registering attributes).

The load increases less than linear with a growing number of users, since duplicate attribute instances per server do not have an impact. A growing number of servers has a linear impact as well as the length of the search field content. Nonlinear growth of load is caused by the privacy preserving negotiation algorithm while calculating the power set of the atomic parts of the search request. This is given by 2^{n-2} (minus two, because of the concatenation of three fields) where n is the number of search fields. We argue that this is not an issue (Figure 5.2), since we expect the number of search fields to be usually small (less than six).

Giving a second example scenario, we assume the profile attribute length, the profile field name length, the sender address length, the host name length and the key length to be 255 bytes (=255 characters) each, which represents the worst case. Based on our prototype, we assume a CHORD DHT to be the lookup service. Furthermore, we assume users to fill three search fields, a fraction of 30% of the servers having at least one user with one matching attribute field and a fraction of 5% of these servers having at least one user hosting a matching user profile. Subsequently, we assume that two users on each server in the candidate list are matching all search fields, and each of these users filled out five profile fields.

Since we assume the maximum field lengths (URL restrictions) in our example and do not expect users to use more than three to five attributes per average search operation in reality, we assume our scenario to be a rather worse case. As a consequence of this scenario evaluation, we argue that the resource consumption is not an obstacle for deploying this protocol and thus the privacy is preserved at an affordable amount of costs. Hence we meet our requirement 2.

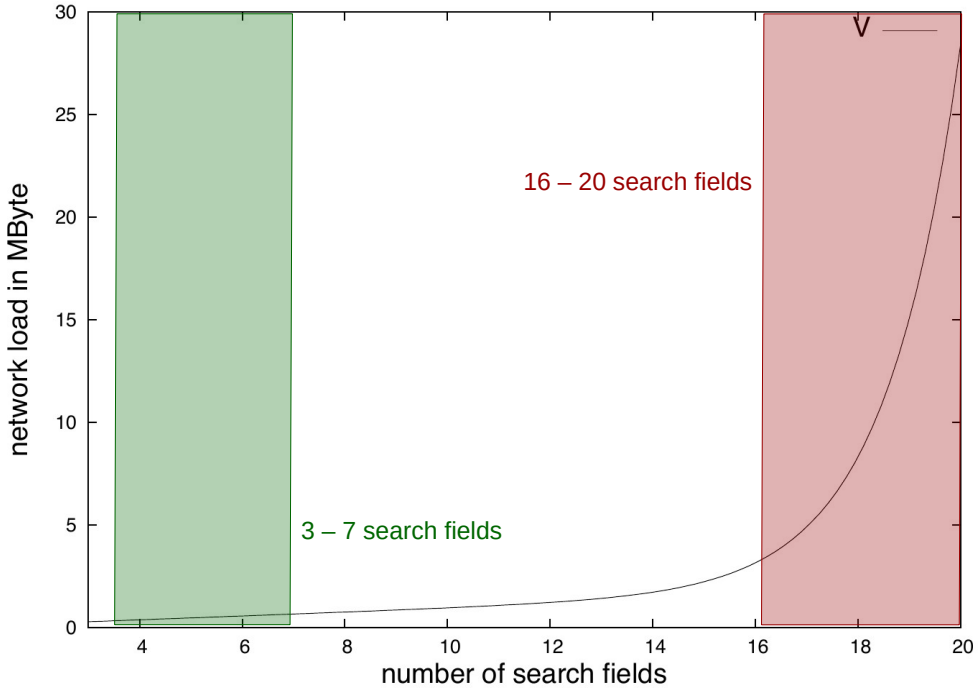


Figure 5.2: Impact of growing number of search fields on our example scenario (1,000 servers)

To prove the functionality of our scheme in a testbed, we implemented a proof-of-concept prototype by adding components to the XMPP server Tigase[6]. The choice was driven by the openness of the latter as well as by the fact that motivating existing XMPP users to help us testing is easier than motivating people to join Diaspora* for that purpose. Creating enough load to the system for exploring its performance limits, was done by a small script for sending random requests.

Beside the result that the idea works well, we learned that the performance is sufficient not to be prohibitive even though we did no performance optimizations in our code and used comparably slow machines (2.2 GHZ AMD cores). Assuming that users may want to find in average a dozen friends in a month, this server setup can handle more than 50,000 users at one core without further code optimizations.

5.5 Related Work

The authors of Vis-a-Vis [Shakimov et al., 2011] suggested a DOSN approach based on virtual individual servers. Every user runs her own virtual server for profile data availability and interaction handling. A hierarchical scheme for finding user handles and group management is proposed, realized by spanning a DHT across the individual servers. Data which is used to characterize users in the search scheme is defined as "searchable" which means to be public. We consider this search scheme to be the closest to our proposed solution but it does does not meet our privacy requirements.

[6] `https://projects.tigase.org/`, accessed on 2015-04-17

To the best of our knowledge, no current work exploits the properties of decentralized server architectures for finding user handles with privacy. Several approaches implement client-server based DOSNs like e.g. Diaspora* and Jabbix[7], but they do not provide any integrated comprehensive user discovery. XMPP addresses can globally be found by using central user directories, based on XEP-0055 (Jabber Search), but there are "no security features or concerns related to this proposal"[8]. Server-local JID (Jabber Identifier) search is possible with the "Net::XMPP::JID - XMPP JID Module" [9] but again comes without tackling privacy and SPAM concerns.

The situation for finding e-mail addresses is even worse. Attempts to provide system-wide search functionalities did not succeed. For example, the leading German telecommunication company took the e-mail directory[10] (public, without privacy protection) offline. Hence, finding e-mail addresses causes the need for side channel communication via e.g. web pages or social networking sites.

Since finding user handles is a subproblem of finding arbitrary resources, general purpose search approaches are feasible for finding user handles as well. Common ground of these approaches (e.g. keyword search [Reynolds and Vahdat, 2003], XPath [Bonifati et al., 2004], ROAR [Raiciu et al., 2009], SplitQuest [Lopes and Ferreira, 2010] or Bubblestorm [Leng, 2012]) is the public-accessibility of the search strings. Thus, these solutions do not meet our requirements 1, 3 and 4. Our solution, in contrast, allows for the privacy preserving publication and discovery of users, which we achieve by two cascaded indirection schemes.

5.6 Conclusion

In this chapter, we presented the first approach that allows to find user handles in systems of decentralized client-server architectures, without disclosing any data that is linked to the permanent user handle (ID or address). The novelty is to avoid building a user-linkable search index. This is realized by cutting the search strings into atomic parts, which are linked to the server handle instead of the user handle and separately registered at the lookup service. The reunification of the search results is done locally and the subsequent access to a temporal user handle is limited to requests, which demonstrate a minimum knowledge about the search target. The permanent ID (e.g. email address) is not published by this mechanisms until the user is manually responding to the request and thus confirming the contact to be desired.

It hence renders offline or other side channel communication for the purpose of discovering users unnecessary and thus increases the usability of email, or Jabber-like IM services. It additionally represents a step towards creating a usable distributed social networking service, based on decentralized servers since finding friends is a core functionality in today's OSNs.

[7] https://jabbix.com/, accessed on 2014-02-18
[8] http://xmpp.org/extensions/xep-0055.html, accessed on 2014-02-18
[9] http://search.cpan.org/dist/Net-XMPP/lib/Net/XMPP/JID.pm, accessed on 2014-02-18
[10] http://www.email-verzeichnis.de/, accessed on 2014-02-18

Our extensive evaluation includes the functionality, the privacy and security as well as the communication costs, coming with the usage of our approach. A prototype implementation based on XMPP, extending the popular Jabber server Tigase, underlines feasibility and scalability of the system.

Chapter 6

Increasing Profile Availability in P2P-based OSNs

Doing away with a centralized service provider, and realizing social networking features over a decentralized user-contributed (P2P) infrastructure is a promising approach, that has been proposed and explored for over half of a decade [Buchegger et al., 2009b, Cutillo et al., 2009b, Sharma and Datta, 2012, Durr et al., 2012, Nilizadeh et al., 2012, Aiello and Ruffo, 2012, Jahid et al., 2012]. Most existing works aim to emulate full-fledged features of popular commercial social networking services like Facebook and Twitter. Despite many academic as well as open-source community initiatives, P2P OSNs are still not sufficiently mature to provide an easy to use and reliable service. A major challenge that is still not adequately solved is to efficiently replicate user profiles in case of assuming short session durations (Section 2.2).

To that end, we argue that bulk data storage as a result of sharing e.g. large photo albums should be distinguished from realizing other features in developing a P2P-OSN. This keeps user profiles small and hence easier to replicate. The rationale behind this design choice is that social updates are the glue that binds a social network by keeping it interesting for its users. It is thus important to provide fast dissemination and high availability of such information. In contrast, stale bulk data provides less benefit per data volume and may be abdicable. We thus envision a lightweight P2P online social network, aiming to securely store and disseminate social updates (this includes all types of 1:1 and 1:n communications) over a P2P back-end, but without the burden of permanently storing bulk data such as huge photo albums or videos.

The total volume of such user profiles can be reasonably small (e.g. 1-10MB), and hence can be more easily maintained over a P2P infrastructure. Furthermore, free riding incentives are mitigated by decreasing the size of objects to be replicated despite the fact that members in the network may be individually unreliable.

In case that users still want to keep bulk data, e.g., videos and large photo albums, it could either be stored at third party services, or by peers dedicating more resources, or by realizing a hybrid infrastructure. Thus, a more privacy-concerned user may use

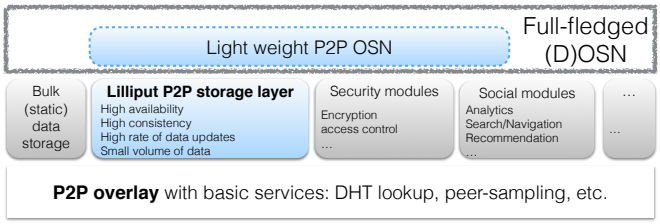


Figure 6.1: Lilliput is a storage module to achieve high availability and consistency for small volume of aggregate storage, comprising of small data items. The latter however can be inserted and updated at high rate in the system. Lilliput's purpose is to store and share the 'social glue' of online social networks such as status updates, profile, messages and text posts on walls, etc., disentangling the storage of (relatively static) bulk data - photo albums and multimedia.

a peers only storage functionality, and pay for it by dedicating more resources herself (e.g., by running a more reliable server, similar to a Pod of Diaspora, to store and serve the bulk of her data), while, a less discerning user may be satisfied with using a cloud service like Dropbox, Flickr or Youtube for media storage. Multi-cloud storage modules such as Depsky [Bessani et al., 2011] or InterCloudRAIDer [Ling and Datta, 2014] could provide an intermediate amount of privacy, by further dividing the bulk data over several cloud services instead of storing them at a single one.

The bottom-line of disentangling the bulk data storage and dissemination from the essential social updates is that the quality of service for the latter is not sacrificed due to the burden of bulk data, while different users can then append diverse kinds of solutions according to their privacy needs and concerns, on top of this lightweight P2P online social networking service. For the rest of this chapter, we will thus focus only on the design of a lightweight, reliable and secure P2P storage substrate aimed at supporting a lightweight online social network. Figure 6.1 showcases how we envision a full-fledged system may look like, and how the other components would be related to the storage service we have designed.

We next enumerate the requirements that we believe that a lightweight online social network should ideally meet. We distinguish between functional and non-functional requirements.

The functional requirements are inspired by current popular OSN functionalities:

1. 1:1 as well as 1:n communication mechanisms for users to communicate (messages, wall, ...)
2. Content sharing functionalities along with flexible access control mechanisms
3. A content and user handle search mechanism
4. Event notifications

The underlying system needs to achieve the following non-functional requirements in order to realize high quality of service while fulfilling the above functional requirements:

1. 24/7 availability of each user's profile data: This includes an inbox for messages, technical information (header data) as well as posts and thumbnails of photos and videos.
2. Consistency guarantees for the profile data: Users need to be sure that the received data is correct and not stale.
3. Fast access to every profile, with propagation delays comparable to similar web-based systems.
4. Feasible resource consumption w.r.t. what is available from peers in terms of storage, bandwidth and computational power.
5. Cryptographic access control and authorization mechanisms providing confidentiality guarantees.

In this chapter, we present a design of a storage service which uses peer resources instead of any dedicated infrastructure, advocating a redundancy management scheme which tries to identify a sweet-spot in the trade-offs between high profile availability and low resource consumption. To that end, we present Lilliput, a set of protocols to realize a storage service for lightweight P2P online social networks, specifically achieving the following:

- Up to 99.64% of user profile availability even under churn levels where the session durations exhibit a median of 50 minutes and online peer population percentages are between 7.9% and 15.5% of all nodes.
- This very high level of availability is achieved without overstretching (scarce) network resources, e.g., the average transfer rate of the highest contributors (10% fraction) is 68 kbit/s and respectively 44 kbit/s for the lowest contributors (90% fraction).
- Despite providing high availability, Lilliput, based on a very small but aggressively maintained set of replicas, avoids running into the consistency gap typically occurring in replication systems.

One can notice the conflicting system requirements - high availability of user profile data has to be achieved despite using (extremely) unreliable user devices. Placing many replicas of each piece of content in the network could arguably achieve this, but that would be detrimental to the network resource utilization, particularly given the high rate of updates under which consistency needs to be maintained.

Lilliput handles these conflicting requirements by leveraging small data overlays with dynamic participation of nodes. These data overlays can be created by any user who will be regarded as the data owner for that data overlay. Subsequently, other nodes are invited to her data overlay with the help of the members of the overlay. Once a sufficient number of nodes have been invited to the overlay, it typically becomes self-supporting. At this point the data owner can disconnect and the other participants in the overlay will continue to invite nodes upon necessity. Nodes in the overlays maintain knowledge about the current replica status. Each node is aware of the status of every other participant in the overlay. Based on this knowledge, the set of nodes which replicate one profile can react to critical situations by inviting new nodes.

In our packet based simulation driven evaluation, we show that even small replica sets can successfully keep a user profile online using an aggressive maintenance of a small set of live nodes in the overlay, and can do so by utilizing a small amount of network resources. That is true even under very adversarial churn patterns. It is worth stressing that Lilliput's design allows nodes with short online durations to contribute, avoiding unavailabilities of user profiles without overloading relatively highly available peers.

The contributions of this chapter are:

- We present a new architecture in the field of P2P OSNs with the focus on user profile availability. Motivated by Schneider et al. [Schneider et al., 2009], who discovered that OSN session durations are short (mean time: 40 minutes), it allows every node in the network that is online for at least a couple of minutes to be able to contribute in keeping data items available by applying our dynamic replication scheme.
- Caused by plenty of data copy processes while applying our protocols, replication of big data items becomes too expensive. We thus propose to reduce the profile size by avoiding bulk content data to be permanently part of the user profile.
- In contrast to existing literature in the field of DOSN, we run packet level simulations to evaluate our system under high churn which exhibits a median session duration of 50 minutes.

The remainder of this chapter is structured as follows. Section 6.1 contains a system design description that details the system environment and specifies data structures and protocols. We evaluate Lilliput in Section 6.2. Furthermore, we discuss related work in Section 6.3 where we point out the void that Lilliput fills, and how this work contributes to the progress of DOSNs. We conclude this chapter in Section 6.4.

6.1 System Design

This section first gives a brief overview of Lilliput and describes the assumed system environment. It further explains Lilliput's architecture in detail, including its components and protocols.

6.1.1 System Environment and Brief System Overview

The system environment is determined by the user behavior that can be observed in OSNs as well as by today's technical preconditions with respect to network bandwidth and user's devices.

The user behavior has been investigated in Chapter 2. The main findings are that both - the session durations and total online times of users - are short. Finding a small and static subset of nodes to replicate user profiles hence seems unfeasible. Moreover, users view very fresh contents that very often consist of shared (often external) links, likes and comments. The popularity of profiles is assumed to be disparate. Users may have just a couple or even plenty of friends.

Based on today's technical preconditions for accessing OSNs, we assume Lilliput to be run on a vast variety of different devices. This includes desktop machines, laptops, tablets as well as smart phones. From the resource's point of view, it implies that the potential to contribute storage, computational and bandwidth resources to the OSN is heterogeneous. Our strategy to handle this situation is to keep the resource requirements low enough to allow all nodes to participate equally to achieve fairness. The user devices that contribute to the OSN are assumed not be accessible all the time. Instead, we assume users to connect their devices to the network at the moment when they gain a benefit from its web applications.

We handle this challenging situation by introducing an agile replication scheme while strictly limiting the size of user profiles. We further minimize the data replication overhead by leveraging nodes that rejoin overlays which they have been part before, still storing a recent copy.

Participants who join the social networking service for the first time create their profiles and establish one Lilliput overlay for its management. The Lilliput overlays are small, fully connected, and are uniquely identified by IDs that are derived from the handles of the users who created them. They consist of nodes that replicate the respective user profile to keep it available and accessible. The limited size of only three to nine nodes helps achieving scalability, as it allows for flooding the status of profile and updates internally at affordable cost. The exact size is adapted according to system settings and environment, below.

Each Lilliput overlay is registered and can subsequently be found using a discovery service. The discovery is not part of this approach and can be implemented using any of the well known approaches (DHT, DNS, central registries, etc). This choice is orthogonal to Lilliput, and the impact with respect to our study is negligible.

6.1.2 Definition of Data Structures

This subsection details data structures that are established and maintained for the operation of Lilliput.

User Profile

A user profile is assumed to be a container, enclosing all data items owned by one user. It is consisting of a header for meta information (size, IDs, ...), a payload section containing data to be downloaded upon interest as well as an inbox to leave messages addressed to the profile owner. Integrating all data items of one user into one container reduces the effort to check the integrity and simplifies replica maintenance.

We suggest to apply the PMS-SK scheme in [Günther et al., 2012] to protect the confidentiality of user data. This profile management scheme provably protects user data from unintended access as well as perfect unlinkability. Only legitimate users themselves know whether they are allowed to access a certain piece of information. We also use the suggested key handling and the provided operations on the user profiles.

As it is proposed by Günther et al. in [Günther et al., 2012], we define all data items to be stored in key-value pairs: "A profile P is modeled as a set of pairs $(a, \bar{d}) \in \mathscr{A} \times$

$\{0,1\}^*$ where $\mathscr{I} \subseteq \{0,1\}^*$ is the set of possible attribute indices a and d corresponding values stored in P. We assume that within a profile P attribute indices are unique. Furthermore, we assume that each profile P is publicly accessible but is distributed in an authentic manner by its owner $U_P \in U$. Also, every user U owns at most one profile and the profile owned by U is denoted P_U."

PMS does not offer an easy way to implement messaging. To solve this issue, we suggest to allow everyone to include one encrypted link into a message receiver's public section of the profile. This link points to the message (or a set of messages) which is stored at the sender's profile to mitigate the chances to overwhelm the receiver's abilities to receive messages. Every link is signed by the message sender and the profile replication node which accepted to include the link into the public section of the profile.

Since only one link, which can point to a set of messages, is allowed per sender, thousands of senders can drop message notification links in a single profile. In the unlikely case that a storage limitation is reached, we suggest to proceed like it is done in the current e-mail ecosystem and refuse to add new links. The profile owner needs to make sure to empty the space after noticing the messages. To avoid SPAM, a proof of work mechanism might be used in future to legitimate nodes to add the notification link to other user's profiles.

Candidate List

The candidate list is a list of IDs of nodes which are considered to be invited to Lilliput overlays in case of insufficient overlay size. The candidate list contains nodes encountered during normal operation. This includes nodes which have been encountered in any of the joined Lilliput overlays, during profile requests, or which have been obtained from the discovery service. The candidate lists are shared among all participants in all Lilliput overlays. To reduce the number of stale entries in the candidate list, each node removes candidates from its list when the candidate has denied an invitation request or in case of timeout.

6.1.3 Bootstrapping and Maintenance Protocols

This section contains all protocols that are necessary to establish and maintain data overlays in Lilliput. We assume the existence of a P2P overlay with basic services like DHT lookup and peer-sampling.

Bootstrapping

Each participant first chooses an identifier, which is used for identifying both the node in overlay operations and the profile within the social networking service. These IDs are either generated by hashing a unique string, like an e-mail address, or chosen at random. The further bootstrapping process is twofold: First, the node joins and registers within the discovery service. It then creates a profile and the corresponding Lilliput overlay, establishes an initial candidate list, and resumes normal operation.

Creation of Data Overlays

In order to maintain profiles in Lilliput, a data owner has to establish her own data overlay in the system. First, the application locally creates an initial data structure containing the overlay ID and initial application defined data. The ID of the created data overlay equals the node's own ID. It then leverages the lookup service to randomly select the allowed minimum number of nodes (r_min) to invite. When nodes accept the invitation, the owner establishes a TCP connection to send the initial data structure. This procedure is repeated until r_min nodes have accepted and received the initial data structure. The owner will then become the initial leader of the overlay until her node goes offline.

Invitation Procedure

At every point in time at least one node of each data overlay has to be available. Assuming high churn rates, it becomes unlikely that the initial set of nodes selected by the data owner is able to achieve 24/7 availability. Therefore, the nodes in the data overlay will select and invite other nodes whenever too many nodes become unavailable.

The invitation process for each data overlay is led by the data overlay leader. The data owner will always be the leader of her own data overlay as long as her node is online. In the absence of the data owner, the node whose ID is closest to the overlay ID is elected using a form of the Bully algorithm. Since every node in the data overlay knows the overlay IDs of all other nodes, the first node to detect a failure will calculate the distance between the overlay ID and its own ID as well as the distance between the IDs of other overlay members and the overlay ID. Heartbeat messages are then sent to elect the invitation leader. Please note that since the owner has the same ID as her overlay, once she is online, she always wins the leader election.

If necessary (number of nodes in the overlay drops below r_min), the leader of an overlay selects a node from the list of candidate nodes taking a combination of metrics into account. The leader will then send an invitation message to the candidate node. The invitation message contains the overlay ID, the leader node ID as well as optionally the ID and IP of another node in the data overlay with which to perform the initial data transfer.

The invited node can either respond with an invitation refusal or with an invitation acknowledgement. Invitation refusal, e.g. due to exceeding the maximum storage on a node, will result in the node being removed from the leader's candidate set and the next candidate will be invited. However, if the node accepts the invitation, it will then open a TCP connection either to the leader or to the designated download node, if available, and request the overlay's data.

After a successful initial transfer, the invited node sends a data acknowledgement to the leader as well as the first heartbeat messages to all other nodes. After that, the node is considered an established participant in the data overlay.

Monitoring the Overlay Status Information

The idea of dynamically inviting nodes to join the overlays upon demand causes the need to monitor the number of available replicas (the current size of the overlay).

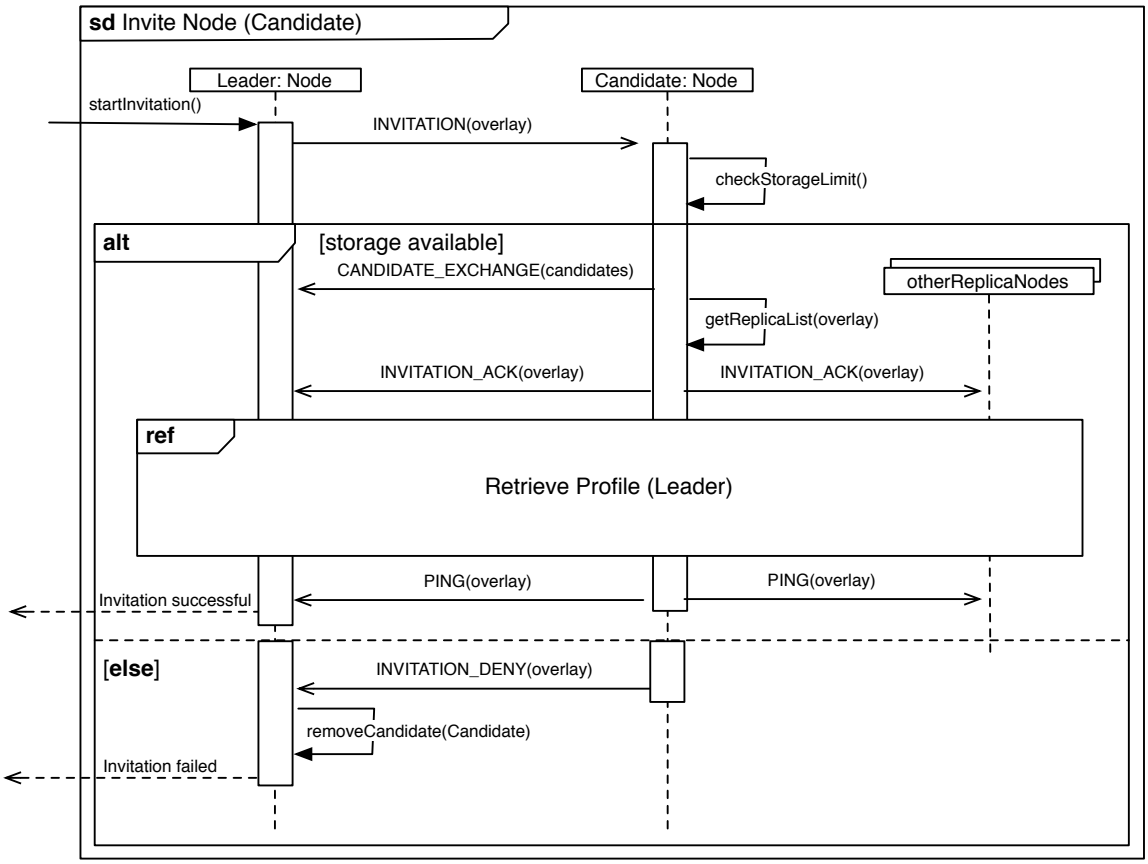


Figure 6.2: Sequence diagram of the invitation process

Thus, heartbeat messages are periodically exchanged UDP messages among nodes in same data overlay. They have two main purposes: First, they are used to detect failed or disconnected nodes in order to achieve a constant awareness of the number of nodes in the data overlay. Second, they are used to disseminate and measure other information about nodes, such as current utilization of resources or average round trip time (RTT) between nodes.

Reconnection of Nodes to the Overlay

To reduce bandwidth consumption, nodes that come back online after a downtime will first try to reconnect to the data overlays which it was previously participating. In case the reconnect fails, the node will - depending on the local storage limitations - eventually discard the data to free storage for other data overlays. This is not the case for the data owner's node in case of reconnecting to her own data overlay. Instead, she will re-instantiate her data overlay in case her node can not reconnect.

6.1.4 Application Protocols

The category of application protocols encompasses protocols used for read and write access of the data, which is stored in overlays. The receiver in these protocols is always

a node that currently participates in the overlay while the initiator is the application that is using the system for its service.

Access to Data Overlays

Three operations are supported by the data overlays. Read operations incorporate data requests either for the whole data, for parts of the data (e.g. delta updates) or sections of the data structure. Update operations performed on the private section require owner's rights (cryptographically enforced). Write operations targeted to the public inbox are strongly limited in size and frequency per client and are non-revocable. Public inbox write operations are facilitating asynchronous message exchange among users as well as push notifications to inform users about profile updates.

Data Dissemination

Write operations (see Access to Data Overlays) are always performed between the client node and only one node from the data overlay. The contacted node will have to take care of disseminating the data to the other nodes, which is done in an iterative way. Once all nodes in the data overlay have acknowledged the write request, the first contacted node sends an acknowledgment back to the client. Updates by the owner are handled in a different way, since the owner's node will always participate in her own data overlay for the duration of the node's online time (see below). The owner node will simply take the application request and perform the iterative updates itself.

6.1.5 Node Selection Strategies

The invitation leader has to choose one node from the candidate list to perform the invitation process. However, the candidate list contains more than one node during normal operation. The strategy to choose the candidate to invite has an impact on the performance of the system. We suggest the following strategies:

- *random:* nodes are selected from the candidate list at random
- *equalizeConnections:* A node could be in replica overlays where all sets of replica nodes are disjunct. Taking the maintenance effort perspective, this is the worse case since nodes sharing more then one overlay still only have to ping each other once in each interval to check if the other is still available. However, the opposite case where exactly the same group of nodes hosts multiple overlays is the worst case from the stability point of view since one failing node causes invitations in each affected overlay.

  To strike a balance between both extremes, we propose to modify the node selection in such a way that each node tries to equalize the number of shared overlays among the nodes which it is currently sharing overlays with. A candidate that is in only one other overlay together with the inviting node should therefore be preferred over a node that is already in many shared overlays. To do that, the node selection algorithm calculates a score for each candidate that is inversely proportional to the number of shared replica overlays s. The node with the highest score is then selected for invitation.

A special case ($s = 0$) is necessary because otherwise candidates that are not yet in any shared overlay would simultaneously get the highest score in every candidate list in which the node is part of. To avoid this case, new nodes' score is a random value between 0 and the maximum achievable score (s_{max}). The small random factor ($rand([-0.01, 0.01])$) ensures nodes with the same score to be chosen randomly:

$$score = \begin{cases} s_{max} \cdot rand([0,1]) & \text{if } s = 0 \\ (s_{max} - s) + rand([-0.01, 0.01]) & \text{if } s \geq 0 \end{cases}$$

- *filterShortTimeThenEqualizeConnections:* Using this node selection strategy, nodes are scored according to equalizeConnections while omitting nodes that have been online for less then two minutes. The background is twofold: First, nodes coming online for shorter than two minutes cannot contribute much to increase profile availability but cause synchronization traffic. Second, users may want to keep their own network link utilization low in the first seconds to retrieve updates of own interests first.

6.2 Evaluation

To assess both the efficiency of our system as well as the availability it can provide, we performed an extensive simulation study. Addressing the requirements to a lightweight DOSN, we focused on the following questions:

1. At what fraction of time a profile and its respective overlay are available, and how many profiles are offline at any given point of time during the study?
2. How much data in total is being transferred during the entire simulation and what fraction of the bandwidth per node is used?
3. How many overlays does each node have to join such that Lilliput can achieve an acceptable level of profile availability?

We now briefly describe the assumptions, the simulation environment and the results of our simulations.

6.2.1 Churn Assumption

Churn describes how peers arrive at and depart from the system over time and reflects the unreliability of nodes in P2P-based systems. Measuring the performance of P2P-based systems is always related to the churn model, since the churn model determines the availability of resources in the network.

In the simplest case, churn can be generated from a single probability distribution. To generate churn for n nodes, a delay time is drawn for each node from a population (following a certain distribution), after which the node is brought online. Subsequently after determining the time a node is appearing, a session duration is defined.

However, studies of churn and user behavior [Benevenuto et al., 2009, Steiner et al., 2009] as well as our own study in Chapter 2 show that the simplest methods to generate churn do not reflect the situation in reality. Just drawing both the online duration and the inter session time does not create diurnal patterns and time zone distributions of nodes in the world as well as week day related patterns.

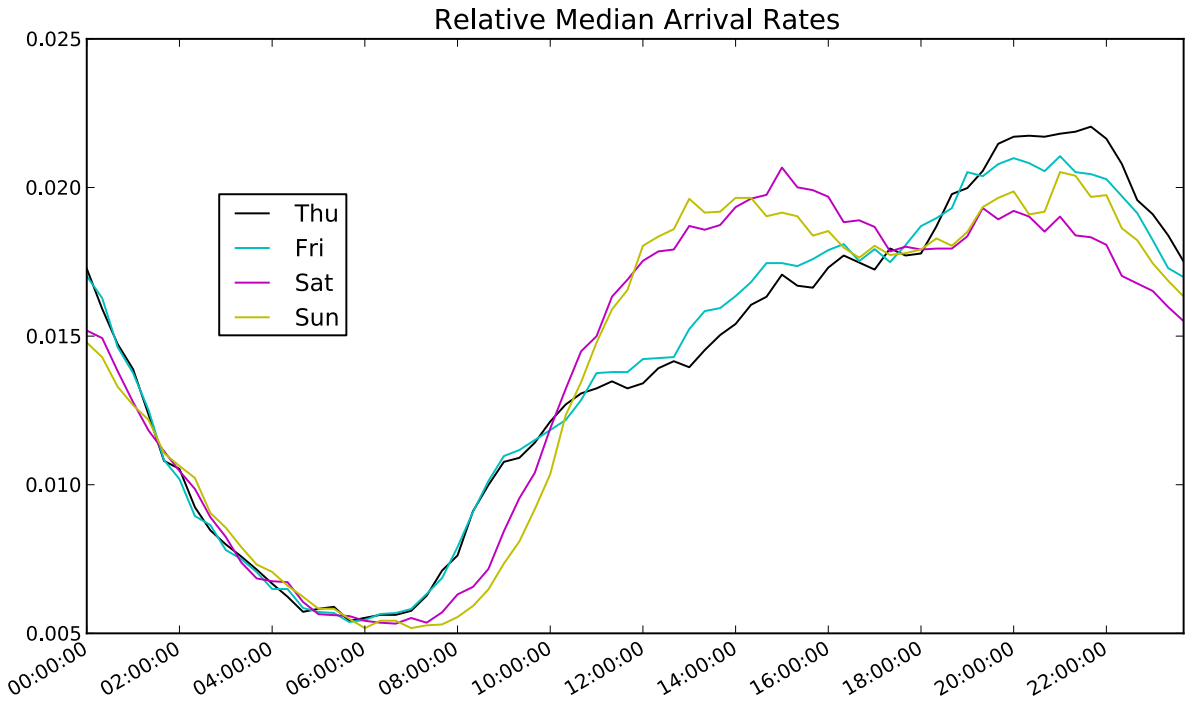


Figure 6.3: Diurnal patterns of churn with respect to weekdays; Monday till Wednesday equal Thursday and have thus been omitted in favor of lucidity

While our own user behavior study (FPA) focuses on the orchestration of functionality and is up-to-date, it is not ideal to build churn models. The reasons are that:

- our participants are originated from 46 countries but only the German subset has a reasonable size,
- a subset of our participants does not share the country / timezone information with us.

Thus, we rely on existing churn models from the literature that are in-line with our results to evaluate Lilliput. The KAD trace by Steiner et al. [Steiner et al., 2009, Steiner et al., 2007] and the Skype super-node trace [Guha et al., 2006] are available online[1]. For the following evaluations, we use the KAD trace, since the Skype super-peer trace just contains the most stable nodes in the system. It hence is too optimistic to evaluate a P2P-OSN. We derived arrival rates for single timezones that exhibit a clear diurnal pattern. Our trace generator, based on these traces, generates synthetic trace files for arbitrary numbers of nodes, simulation durations as well as timezone distributions of nodes.

[1] `http://www.cs.illinois.edu/~pbg/availability/`, accessed on 2015-11-01

Please note that the KAD trace is more challenging than all traces or churn models which have been used to evaluate P2P OSNs before. We assume in average about 11% of nodes to be online and have situations with just 7.9% of nodes in the system.

To be able to simulate different network sizes as well as to avoid to use a part of the trace recorded during an uncommon situation in the observed system, we build a churn generator that generates trace files with the desired properties from the KAD trace. This churn generator is available online[2].

6.2.2 Simulation Environment and Experiment Setup

In this section, we name the simulation environment and describe the modifications on it that were necessary to run our simulations. Thereafter, we specify the setup of our experiments.

Simulation Environment

We used the general purpose OMNeT++ network simulator [Varga and Hornig, 2008] in conjunction with the OverSim overlay network simulation framework [Baumgart et al., 2007]. OverSim supports simulations driven by trace files. However, we needed to make modifications to OverSim in order to allow nodes to leave the system and return later with their state preserved. Furthermore, we modified OverSim's SimpleUnderlay network topology so that we could control the latency between nodes based on their assigned timezone. For this purpose, nodes are placed into an euclidean coordinate system according to their timezone.

Setup and Parameters

We simulated the system using several simulations sizes and scenarios using 1000, 5000, 10,000 and 15,000 nodes as well as four different timezone distributions for these nodes. We simulated the nodes being located in one country, one continent, two continents or around the world using a distribution taken from the data collected by Pingdom[3]. Each trace spans five days of simulation time.

Nodes in the simulation are considered homogeneous having a reliable Internet connection (1MBit uplink, 10MBit downlink) as well as 500MB of available storage space. Latency between two nodes is calculated by the SimpleUnderlay topology from OverSim, which uses a fixed value of 20ms and adds a fraction of the euclidean distance between the position of two nodes as well as some random jitter, the maximum latency is approximately 600ms.

The system parameters that are elaborated in our evaluation influence the size of the overlays, the maximum profile size as well as the time until a node starts reconnecting to the overlays:

[2] https://www.p2p.tu-darmstadt.de/research/p2p-churn-generator/, accessed on 2015-11-01

[3] `http://royal.pingdom.com/2013/02/12/internet-users-time-zone/`, accessed on 2015-11-01

r_min

This value defines the lowest number of nodes which is acceptable to have in one overlay and must be set to a value ≥ 2 to ensure that each profile is replicated. We evaluated values between 2 and 4 since values bigger than 4 did not help to improve availability but caused resource consumption.

r_max

The valid value range for this upper bound for the profile size is between $r\_min + 1$ and ∞. If r_max is set to ∞, returning nodes are never rejected to rejoin the overlay when reappearing online. In all other cases, nodes are rejected when the replica count of the overlay grows up to r_max. We decided to evaluate all combinations with $r\_max \leq 9$ since a growing overlay size makes flooding of heartbeat messages expensive.

profileSize

The size of profiles is limited to 10 MiB. We decided to use this profile size for two reasons: downloading our own profiles in Facebook resulted in a data package of less than 10 MB and Lilliput is not designed for long term storage of bulk data. We thus argue that this profile size is appropriate.

maxStorageSize

The device owner may want to limit the storage which is used by the system to host other users' profiles. We evaluated the effect of this limit on profile availability and network load for the values 500 MiB, 1000 MiB and ∞.

Measurements

From the 14 days of simulation time, the first five days are used as warmup period while we observe the key metrics of the system during the remaining days of simulation time. We measure the consumed storage space, the bandwidth utilization, the number of connections as well as the number of overlays each node participates in. For the data overlays, we observe the overall availability over time, the number of concurrent active nodes and the total number of nodes invited during the course of the simulation. Finally we measure the ratio of data overlays available with regards to the total number of created overlays over time.

6.2.3 Results

We provide the results of our experiments in the following section. As a first step we justify the size of the experiment and show that increasing the number of nodes by factor 15 does not change the results in general. The main focus of the numerical results in this section covers the achieved profile availability, the communication costs as well as the storage utilization on the nodes.

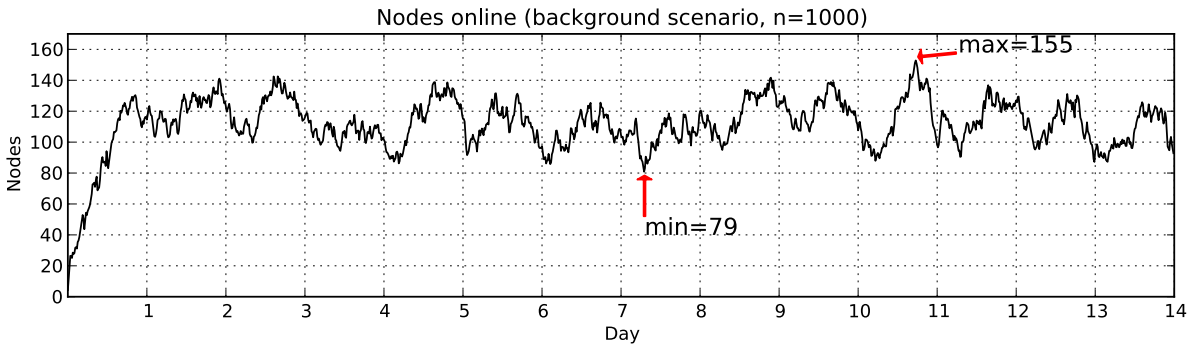


Figure 6.4: Example churn model: The numbers of online nodes in 5k, 10k and 15k churn models are (proportionally) identical.

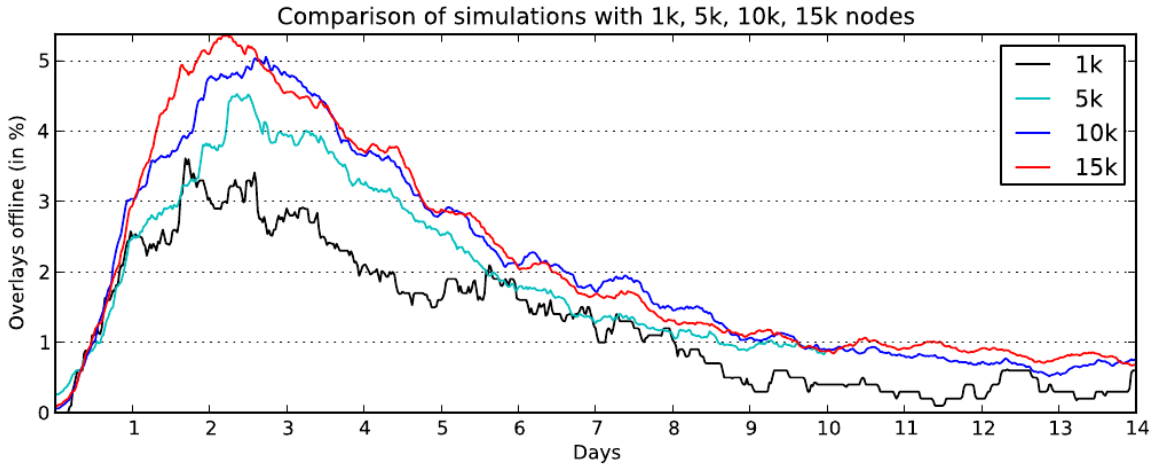


Figure 6.5: Scaling up the experiment: influence of the total network size on the availability of profiles

Profile Availability

Not considering the bootstrapping phase, we discovered parameter combinations which led to an availability of more than 99.07%. That is especially true for (r_min/r_max) 3/6, 4/5 and 3/7. In case of 3/7, 96.37% of the profiles have never been offline during the whole simulation time. The profiles which have been offline at least once during simulation time still have a median availability of 84.28%.

The main factors that influence the availability in our simulations are the minimum fraction of online nodes during the observation period as well as the time it takes to invite a new node to the overlay. Since we assumed a churn model that results in having a minimum of 7.9% of nodes online (and a maximum of 15.5%, Figure 6.4), we make rather pessimistic assumptions. Please note that the total number of profiles equals 100% of the nodes in the experiments. We do not fulfill our non-functional requirement 1 since we achieve 99.64% instead of 100% availability but consider the availability criteria not to be an obstacle in applying Lilliput.

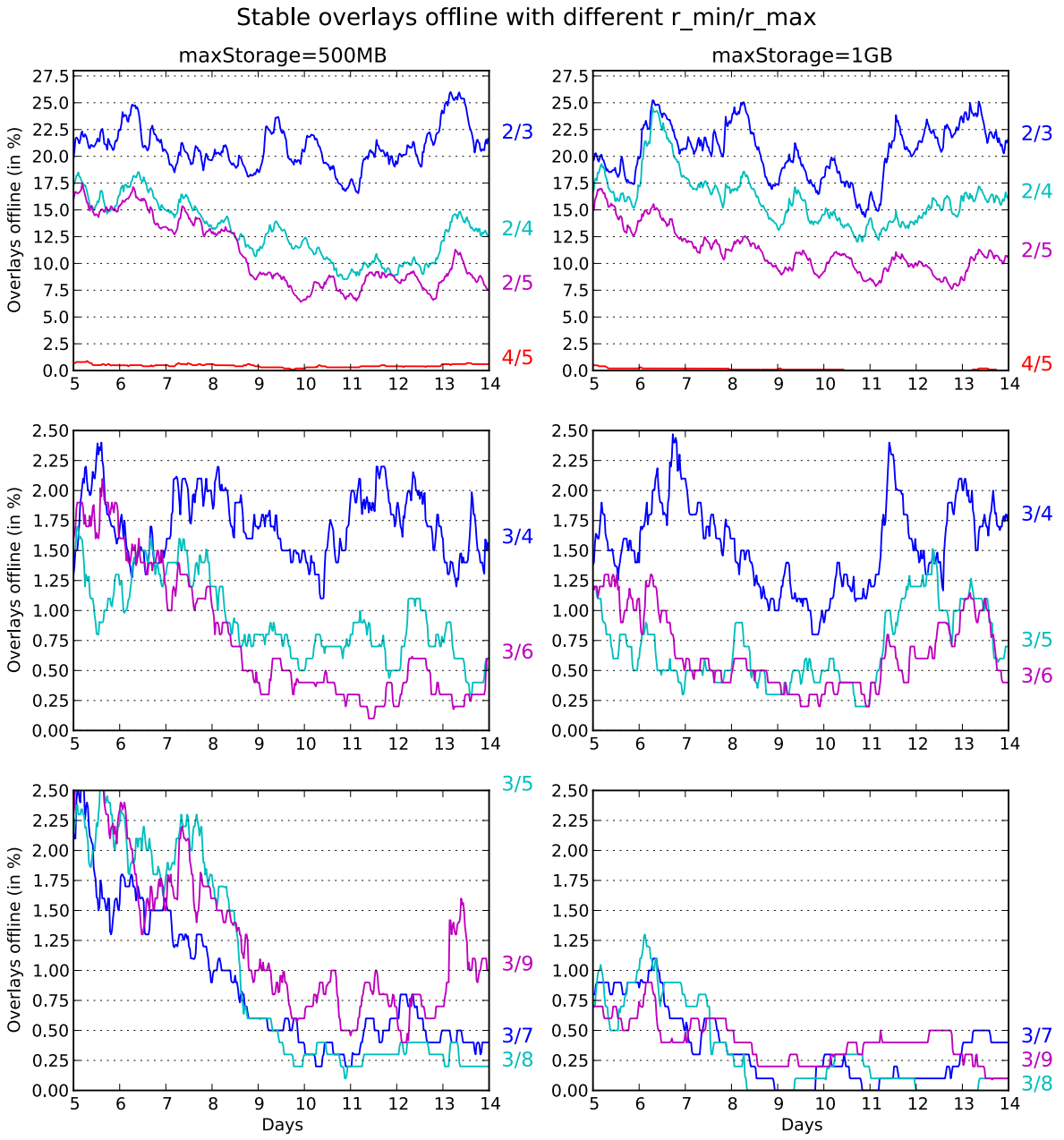


Figure 6.6: Fraction of overlays that have been offline at least once during simulation time under storage limitations with respect to different parameter combinations (r_min, r_max)

# Nodes	# Overlays available 24/7	Total Availability (Time)
1k	97.6	0.9964
10k	96.4	0.9918
15k	96.37	0.9907

Table 6.1: Availability of profile overlays

# Nodes	Mean	Median
1k	0.8515	0.9012
10k	0.7726	0.8627
15k	0.7438	0.8428

Table 6.2: Mean and median availability of nodes that have not been online all the time

Communication Overhead

The communication overhead is a strong concern of users in mobile environments but matters to a certain extent in every case. Users may not want an OSN application to utilize the complete available bandwidth. We hence measured the amount of data which is sent and received by each node. The results show that applying our protocols causes average traffic of about 40 times the profile size per node in a 14 days period.

The bandwidth utilization is driven by the assumed churn, the replication parameters (r_min/r_max), the maximum storage allowance on the nodes and the size of the stored objects. The churn influence on bandwidth utilization can be translated into the probability of nodes to rejoin overlays in relation to those needing a whole new copy.

Figure 6.7 illustrates both the amount of traffic that is necessary to keep 10 MB online for 14 days as well as the influence of the parameters r_min, r_max and the maximum storage capacity. The influence of the parameters r_min and r_max is twofold (Figure 6.7): First, increasing r_min strongly increases the bandwidth and storage utilization. Second, increasing r_max decreases network traffic. The increasing effect (on bandwidth and storage utilization) is caused by a higher number of nodes in the overlays (and thus more invitation processes). A higher r_max increases the probability for nodes to rejoin an overlay and updating a stale copy of a profile is cheaper than submitting a new copy.

Storage Consumption

The storage used by Lilliput strongly depends on the chosen parameters for r_min and r_max. We used three different settings with 500 MB, 1000 MB and unlimited storage capacity. While unlimited storage resources cause unlimited growth of storage usage, 500 MB is a good choice for all cases where r_max is smaller than 9 (Figure 6.6).

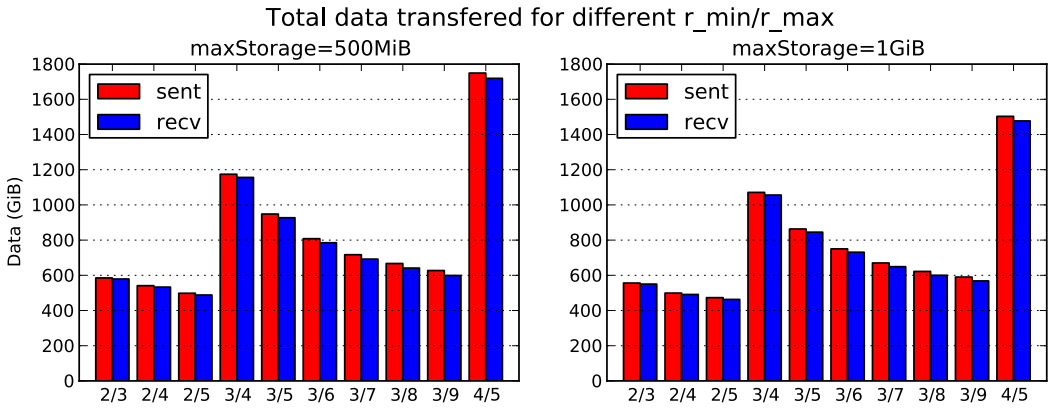


Figure 6.7: Impact of different parameter combinations (r_min, r_max) on the total amount of data transferred in the experiment with 1000 nodes in a 14 days period

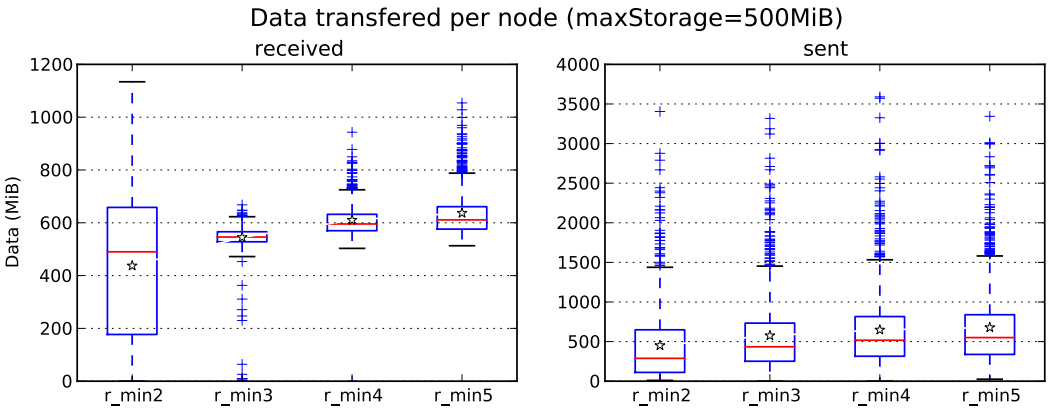


Figure 6.8: Amount of data sent and received per node in the case of limited storage; star-markers indicate the averages; differences between received and sent data amounts are caused by churn-indicated transmission abortions; outliers indicate early / late adopters during bootstrapping phase

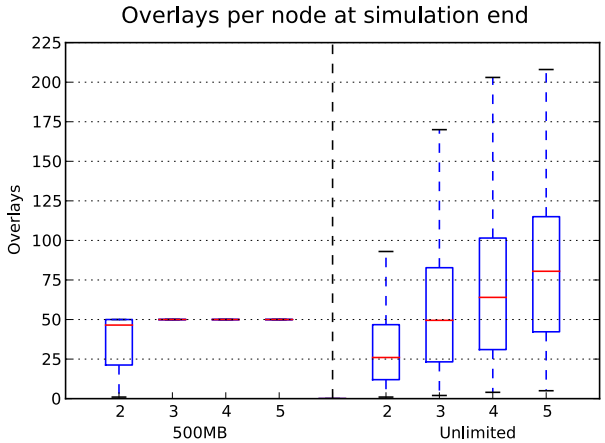


Figure 6.9: Number of overlays each node is part of at the end of simulation time with respect to the minimal overlay size r_min

Data Consistency

Content freshness is not an issue for Lilliput, since the content is replicated in a single, fully connected overlay and the participating nodes share identical copies of the profile. Assuming that the owner of a profile updates the profile, she necessarily is online. If the owner of the profile is online, she serves access requests from the original copy. Before going offline, the owner makes sure that the replica are up to date by uploading her profile to the replicating nodes. The replicating nodes subsequently again serve the latest copy in case the profile owner is not available.

Two cases in which the consistency is challenged remain: First, if a breakdown occurs during the process to update the profile (from profile owner to replicating nodes, or between the replicating nodes). Second, if a breakdown occurs after a message has been left for, but before it has been delivered to the profile owner. We do not see any chance to evaluate or tackle the first issue. Regarding the second one, the only chance for the message sender to make sure that the message arrives is to check back later.

6.2.4 Fulfilling Functional Requirements

Our functional requirement 1 can be met by implementing a push / pull communication strategy. Each user profile contains two sections: a larger one where only the owner has write access as well as an inbox where other users can drop messages and notifications. Text message exchange can be realized by dropping a message directly at the recipient's inbox. In case of sharing content (requirement 2), thumbnails or smaller items can be stored at the sender's profile. To avoid the necessity to check a vast amount of profiles for new content items, an update notification (requirement 4) will be dropped at the inbox of the desired receiver.

In case of synchronous communication where both parties are online, a direct data exchange is realized. The presence information of nodes in the network is implicitly available since profile owners are always part of the overlay and are first responder of any requests.

Finding user handles can be realized straightforward via lookup service (requirement 3). This comes with the disadvantage of disclosing information which is necessary to identify the search target.

6.2.5 Comparison to S-Data

S-Data [Shahriar et al., 2013] is the prior approach that in its features and properties is probably most closely related to Lilliput. S-Data groups all participants, with each node of a group hosting all profiles of all group members. It has been evaluated under a much more conservative churn model, and still achieves a notably lower profile availability. The authors indeed state that "For a mean peer uptime of 8 hours it is possible to have more than 93% of the groups online even under 50% failure rate".

6.3 Related Work

There is a vast amount of related work, with a multitude of focus: general purpose P2P storage systems design as well as P2P-OSN specific works. P2P storage approaches like Oceanstore [Kubiatowicz et al., 2000] or UniStore [Karnstedt et al., 2007] are constructed to provide a persistent storage for predominantly static data. They are not optimized to store social data since consistency of small data objects undergoing frequent updates is expensive to achieve.

Most P2P OSN works take a holistic view on the system in general, and profile availability and dissemination receives peripheral attention [Buchegger et al., 2009b, Cutillo et al., 2009a, Aiello and Ruffo, 2012, Graffi et al., 2008, Jahid et al., 2012, Nilizadeh et al., 2012]. The focus is instead on the system design, the communication protocols and the encryption schemes. In most cases, friends are supposed to replicate profile data or a standard underlying storage service such as a DHT is assumed.

Specialized availability schemes for P2P-OSNs have been suggested in SuperNova [Sharma and Datta, 2012], GemStone [Tegeler et al., 2011], My3 [Narendula et al., 2012] and S-DATA [Shahriar et al., 2013], while profile information dissemination using friend networks has been studied in [Mega et al., 2011]. SuperNova focuses on bootstrapping and incentives, while the focus of [Mega et al., 2011] is purely on dissemination and not on persistent storage, and hence they are not comparable. GemStone, My3 and S-Data are direct competitors from the perspective of a storage service.

The main idea of My3 is to select a subset of friends who are fully trusted to replicate one's profile data and to perform access control. For availability, the scheme relies on the assumption that each user has a small subset of friends whose online time patterns cover the online times of all other friends. This is a patently wrong assumption, and can lead to extremely poor data availability [Sharma et al., 2011]. It also requires users to trust these friends to have complete access to all profile data items and enforce the access control benevolently, and not abuse these meta-information.

In Gemstone, a profile owner chooses a set of nodes based on criteria like "online experience" and "social relation" to determine replica peers. This decision is static in nature during the time when the profile owner herself is unavailable. This leads to two effects: First, the owner's online experience establishes a tendency to prefer nodes with similar churn patterns like the owner. Second, the choice is based on experiences from the past and hence assumes constant and predictable future patterns, and lacks dynamic adaptivity.

S-DATA addresses the latter issue by introducing an external and centralized service with global knowledge to find nodes with complementary churn patterns. Based on this knowledge, groups where each node hosts every profile of all group members are formed. Changes in churn behavior of any single participating node necessitate the creation of new group assignments. Furthermore, users need to trust this centralized service not to misuse the knowledge of churn patterns.

In contrast, Lilliput creates small profile overlays based on the current status of the network without depending on assumptions regarding churn patterns. Thus Lilliput

does not prefer or burden stable nodes for profile replication. Each node can contribute to replica profiles even if the session duration is just a few minutes. Thus, there is no need for nodes to explicitly disclose information about churn patterns and nor is the set of replica nodes limited to the set of friends. The result is a rather agile storage service which nevertheless requires low network resources, while providing highly available yet consistent storage even in the presence of frequent updates in the data, making Lilliput ideal as the storage primitive for lightweight P2P OSNs.

6.4 Conclusion

This chapter addressed the issue of profile availability in P2P-OSNs. The straightforward solution of replicating profile data in a DHT takes the control where to store the own data away from the user profile owner and raises trust and incentive issues. These issues have been addressed by a couple of related works [Sharma and Datta, 2012, Tegeler et al., 2011, Narendula et al., 2012, Shahriar et al., 2013] which suggested group-based replication mechanisms. However, these approaches assume online session lengths which are much longer than those measured in real systems e.g. by Schneider et al. [Schneider et al., 2009]. The owner chooses (based on different criteria) a static set of nodes for replication. This choice is fixed until a new decision is made.

In contrast to the related work, we proposed a dynamic replication scheme that still allows the owner to take influence where to store data. Lilliput preserves the advantage of group-based replication schemes to allow nodes to re-join the replication overlay after a period of offline time. Using packet level simulations, we demonstrate that Lilliput can be deployed even under heavy churn and still maintains data redundancy to achieve an outstandingly high profile availability of more than 99%.

The main flaw of dynamic replication is the higher bandwidth utilization. We thus propose the split design to guarantee small user profile sizes to reduce bandwidth utilization as well as storage overhead. We argued that the main 'social aspects' of an online social network can be achieved in a lightweight manner, disentangled from the storage of bulk data. Accordingly, we designed 'Lilliput', which provides high availability of small amount of data, which can however be frequently updated: essentially corresponding to the profile and notification information in social networking applications.

A lookup service is necessary to locate the data objects in the network. This can be achieved by using a DHT for this purpose. However, resource locators are magnitudes smaller and thus easier to replicate than even lightweight user profiles. The reduced resource requirements caused by small user profiles help to mitigate free riding incentives. Furthermore, our list of candidates for later node invitation consists of nodes that have been met during normal operation. Profile owners can thus prefer to store replicas at nodes with a cooperation history.

Lilliput solves an important challenge of P2P-OSNs to provide profile availability in the presence of high churn. It is thus an approach to avoid implications of today's centralized OSNs since Lilliput contributes to make P2P-OSNs becoming an alternative. However, Lilliput does not support sharing of bulk data such as videos or large photo

albums by design. This is an important flaw that needs to be solved by leveraging alternative services depending on the privacy implication. For example, traditional file sharing could be used to share popular e.g. viral videos and 1:1 file transmissions (supported by presence protocols) for sensitive data. Also, cloud-like services may help to share data.

Chapter 7

Leveraging Locally-available Data to Apply Video Prefetching

The decentralized storage of user data is a consequence of the design goal of DOSNs to store and process data in the user's influence zone (Chapter 4). However, even core functionality of OSNs, such as pull-based newsfeeds [Jahid et al., 2012, Nilizadeh et al., 2012] and finding user handles (Chapter 5), requires gathering of information from different (potentially unreliable) sources. It causes communication overhead amongst nodes and thus delays to collect the relevant data upon request. Hence, DOSNs tend to suffer performance disadvantages compared with today's OSNs. These performance disadvantages are determined by the type of decentralization (e.g. F-OSNs or P2P-OSNs), the data storage granularity, the communication delays and the network bandwidth. P2P-DOSN are the most affected type of DOSN.

We suspect this situation to be caused by DOSNs to focus on privacy and security rather than performance and functionality. However, beside enriching the functionality in DOSNs, more work needs to be invested in improving the performance and user experience to help DOSNs to become a serious alternative. To this end, we examine the user behavior in Facebook with the goal in mind to discover stable usage patterns that allow to predict data requests in future. These predictions can be used to avoid and to reduce request delays.

In addition, these predictions can likely be derived from additional information which is linked to the video. Facebook offers a rich set of metadata such as likes, comments and social graph information for content which is uploaded to the social network. In this chapter, we analyse this metadata and evaluate its feasibility to forecast future media consumption. We set up two user studies to observe video consumption habits of 34 users for 14 days in a mobile setting and 774 users for 34 days in average in a stationary setting. As data gathering for prefetching is a computationally intensive task which can be less attractive on mobile devices, our study is based firstly on stationary devices, using a browser plug-in, and secondly on 34 selected participants who allowed us to track their Facebook usage. We analysed the newsfeeds with respect to the type of entries (text, pictures, videos, links), the pre-click delays, the relation to the author (friend or not) as well as the number of likes and comments.

Nodes in DOSNs technically only have local view on content consumption. We thus relate - in contrast to the previous works - the video consumption of users only with locally-available contextual metadata, associated with the content, to find patterns that can be leveraged to predict future content retrieval. We consider approaches that rely on exchanging content access patterns amongst nodes (users) to be disadvantageous in the field of DOSNs. They potentially harm user's privacy.

Elaborating our measurements indicates that the social closeness of content producer and recipient helps in predicting media consumption for close friends and the family members, which are explicit subsets (groups) of what is called 'friend' in Facebook. We had hoped that prefetching based on affective expressions such as likes or comments can be effective. However, we did not find a strong correlation between the number of likes and comments and the probability of a video for being watched. As a further result, we argue that the time a user spends to evaluate a post before clicking on it can help to decrease startup delays. Our study participants tend to spend much more time to evaluate a wallpost before clicking on it as it would be the case without the intention to click on it. We thus suggest to start downloading videos before a click happens.

This chapter contributes to create a better understanding of predictions for prefetching videos based on likes, events, authorship and timings in OSNs. Applying our strategies can help to reduce network traffic in cellular networks for the mobile and to decrease startup times.

7.1 Background and Related Work

Efficient prefetching strategies are desirable for two reasons: First, they allow to shift network traffic to the most cost and energy effective network interfaces of mobile devices. Second, they allow to reduce video startup delays.

Several studies focus on the impact of startup delays on the perceived QoE. In Krishnan and Sitaraman's work [Krishnan and Sitaraman, 2012] the effect of different video startup times are evaluated. They demonstrate for short video clips such as those on YouTube that an increase of startup time increases the probability that a user cancels the streaming session. Prefetching may solve this issue as content is downloaded before the user requests it, resulting in a significantly reduced startup time.

With the work of Gautam et al. [Gautam et al., 2013], a mobile application is developed that allows to prefetch whole video clips based on arbitrary sources such as social networks or news feeds. The paper focuses on energy cost savings realized when applying prefetching on mobile devices. Zhao et al. [Zhao et al., 2013] demonstrate a custom mobile Facebook application that integrates social network based algorithms for prefetching. It allows the conclusion that social prefetching is beneficial, but hard to conduct.

For scenarios in which video sharing sites are in focus, Khemmarat et al. [Khemmarat et al., 2011] propose prefetching strategies considering related videos, and search query results. In contrast, Cheng et al. [Cheng et al., 2009] present the P2P streaming network NetTube which is customized for the video sharing site YouTube. In a long-term measurement, they show that 99.6% of all videos uploaded to YouTube

have a playback time of less than 12 minutes. They give helpful insights on the structure of a video sharing site and how YouTube's infrastructure could be supported by the P2P paradigm. They propose to prefetch the first part of a video with a fixed length of ten seconds. Both approaches achieve high prediction rates but investigate YouTube, a site which focuses on videos and neglects social ties. Their findings are closely related to caching experiments and recommender systems for video sharing sites.

Online Social Networks are increasingly leveraged by networking and caching researchers to predict media consumption. In [Kaafar et al., 2013], a recommendation-aware content placement strategy for Content Delivery Networks (CDN) is investigated. This work focuses on storing content at different places within the network, whereas prefetching concentrates on downloading content to a device. Additionally Bai et al. [Bai et al., 2013] show social network related caching mechanisms for Facebook as well as in Yahoo News. Caching is powerful, but of course focuses on a large set of users, whereas prefetching through the downloading of content to an individual device can be tailored to each user individually. Prefetching, in contrast to caching in large-scale CDNs, has only a limited view on the OSN data - the view of the user. Thus, future video requests are potentially harder to predict.

An overview on video dissemination in OSNs is given by Li et al. [Li et al., 2013]. They show that popularity of videos has a significant impact on their consumption via social networks. Unpopular video content disappears quickly from OSNs. This work concentrates on shared links of videos on a Chinese OSN that is claimed to be similar to Facebook. In contrast, Li et al. focus on the popularity distribution of videos in general and thus neglect that unpopular videos still can be very interesting for a small subset of users e.g. because they are acquainted with people in the video.

Wang et al. [Wang et al., 2011] investigate the most popular social network in China, called RenRen. They design a P2P-based prefetching algorithm that aims to reduce the video start-up time. The SocialTube [Li et al., 2012] system demonstrates the efficiency of a peer-to-peer-based social network. The purpose of the system is to reduce the playback startup time. The authors show that most of the video views are driven by social relationships and less by interests. We can confirm and refine the results.

However, the related work does not focus on locally-available information to avoid both the necessity of content consumption to be shared with others (for privacy reasons) and the need for trust in other nodes to honestly serve requests. We add new insights on the average pre-click viewing time of each post and its relation to the probability of a video to be watched in near future. The community additionally benefits from practical recommendations to identify the close friends in being a predominant factor of a video being watched.

7.2 Data Description

In this section, we describe the data collecting methods and the data gathered both in the stationary as well as in the mobile setting.

7.2.1 Stationary Setting

Our approach in the stationary setting is to use browser plug-ins for Firefox and Chrome to collect data about user behavior in Facebook. We approached volunteers via newspaper articles to convince them to help us with our research. As a supportive incentive, we built statistic pages that help people to understand their own Facebook usage patterns.

The plug-ins read the Facebook wall of a user while the page was being rendered. Additionally, it collects usage patterns such as clicks. We saved and collected the meta information about which type of content was included in the walls (video, picture, link), the timing information (age of the entry, the time span between displaying and clicking / removing from the screen) as well as whether the author of a content item was part of the friendlist or not. The Facebook user ID, which was collected to distinguish the participants, was anonymized using hash functions to protect the privacy of users.

During our observation period of 123 days, the plug-ins (both versions for Firefox and Chrome together) have been installed by 2071 Facebook users. Since most people did not install the plug-in on the first day of our observation period and since some participants left earlier, the average observation time was 34 days. We observed the phenomenon that many users were extremely passive and clicked only on very few content items. To ensure that we only use valid data, we excluded all cases where less than 100 wall entries were clicked. As a result, our analysis is based on 618,165 wall entries from 774 users.

7.2.2 Mobile Setting

To ensure that our findings from the stationary setting are also valid in mobile environments, we validated our findings by a small, prototypical evaluation on mobile devices. Data of 34 users over a time span of two weeks has been gathered and analyzed. Volunteering participants agreed on anonymously sharing their newsfeed's metadata (likes, comments, etc.), information about their interactions with media posts as well as social graph information with us. We created an app for this purpose. In total, 8370 posts including 742 (8.9%) video posts and 3608 (43.1%) pictures have been analyzed.

7.3 Analysis of the Collected Data

We analyzed our datasets to answer the following questions:

- What type of content can be found in the newsfeed?
- What type of content is consumed?
- Which fraction of the offered videos are watched?
- Does the number of its likes and comments change the probability of videos to be consumed?

	Photo	Video	Link	Total
Total	231,582	46,055	340,528	618,165
Clicked	13,342	5,259	21,636	40,237
% Clicked	5.76	11.42	6.35	6.51

Table 7.1: Summary of the newsfeed entries gathered in our study

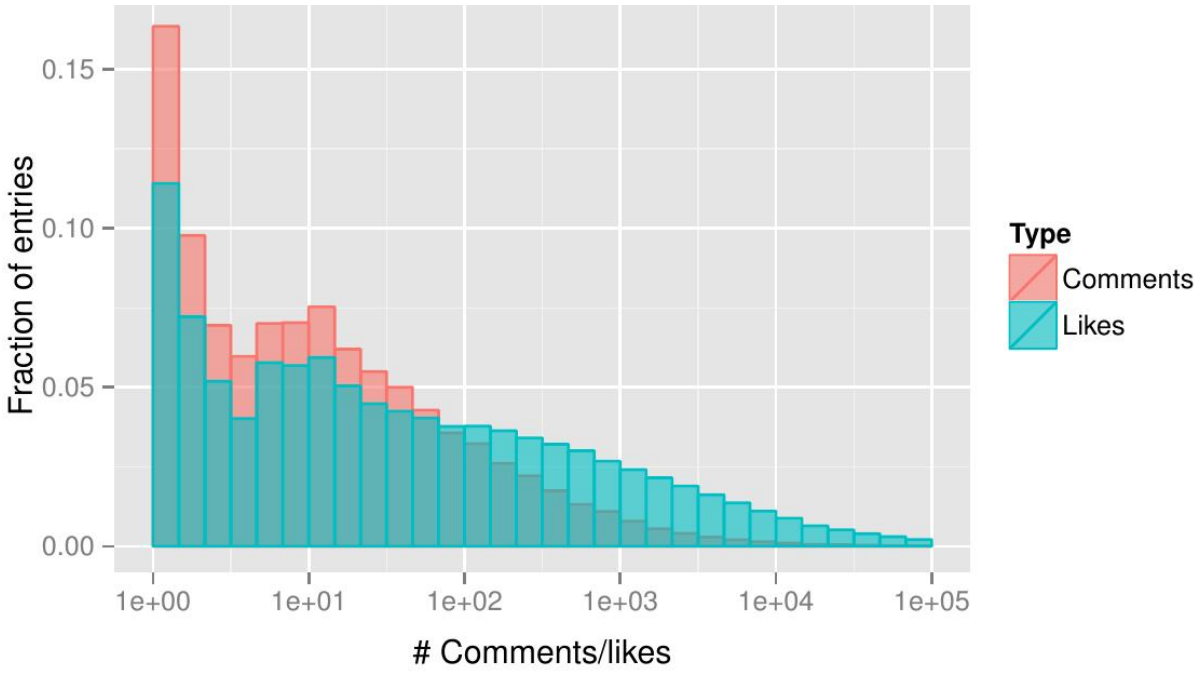


Figure 7.1: Comparison of the number of comments and likes received by newsfeed entries

- Does it depend on the author whether a video is being watched or not? - If yes, to which extent does the type of friendship or the membership in a global group play a role?

7.3.1 Stationary Setting

Our participants viewed an average number of 43 newsfeed entries per day (only days with activity are mentioned). Table 7.1 summarizes the newsfeed compositions with respect to the content type. It also shows what type of content was clicked by our participants.

Interactions with the content are rare in general. Our participants clicked on 7%, liked 4% and commented less than 1% of the displayed content items. Figure 7.1 illustrates the distribution of clicks and likes. Newsfeed items usually have more likes than comments and items with multiple comments are very rare.

We are also interested in the relationship between the author and the user. This can be determined by looking at the friendlist. In average 49.5% of all newsfeed entries are authored by friends. The remaining posts are created by 'pages', (profiles maintained

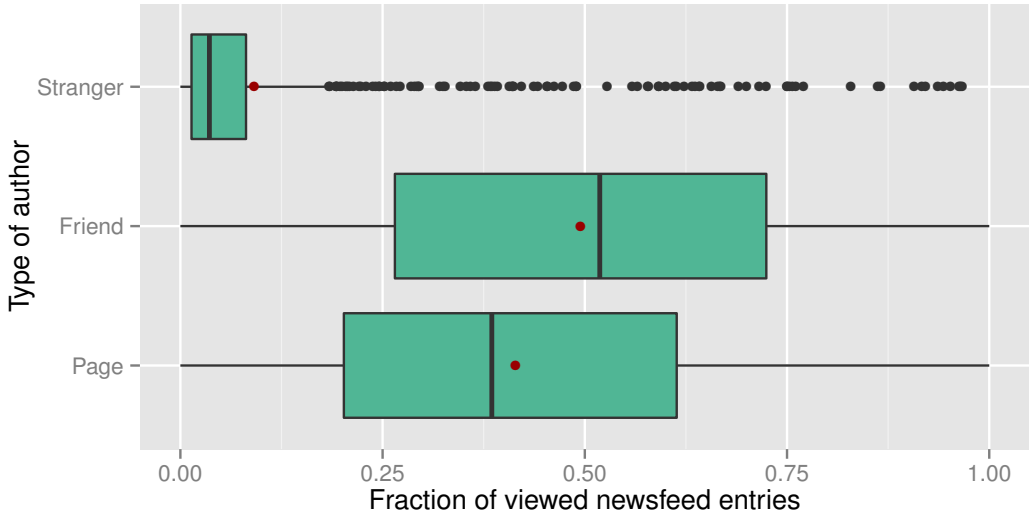


Figure 7.2: Box-Whisker-Plot showing the distribution of clicked content items with respect to authorship; the red dots mark the averages

by companies to spread information) (41.4%) and content from strangers in case that friends liked or commented on these items (9.1%).

Nearly half of the clicked newsfeed entries are those of friends (Figure 7.2). Pages are slightly less popular compared with their fraction in the newsfeed (41%). Nine percent of the viewed content was posted by strangers. Very interesting at this point is that the fractions (with respect to authorship) of watched videos equals the fraction of displayed videos (videos shown on the wall). As a result, authorship cannot be used as a predictor for prefetching content if it is considered at this granularity. We later show in the mobile setting, that authorships of subsets of friends (close friends and family) can be used as predictors.

Figures 7.3 and 7.4 show the distribution of the number of likes and comments, which clicked and non-clicked videos have received before our study participants discovered the vidoes in their newsfeed. In both figures, the distributions are very similar. They cover each other more than 90%, which means that the fraction of watched videos is nearly uncorrelated with the number of likes or comments. If the number of videos with a certain number of comments or likes increases, the number of watched videos with this certain number of comments or likes increases equally and the fraction of watched videos stays the same. That indicates that the number of likes and comments, attached to videos in Facebook, are no feasible prediction basis.

In Figures 7.3 and 7.4, we distinguished between professional pages, maintained by companies, and users since Facebook seems to use different algorithms for choosing newsfeed entries to display them on the user's wall. Our data indicates that content from professional pages needs to have far more likes and comments than items from users to be included into the newsfeeds of users.

However, the timing patterns (Figure 7.5) show a very clear indication that newsfeed entries that will be clicked, are watched for a longer time than those which will be removed without a click. This effect can be used to decrease the startup delay by starting to prefetch the item before it has been clicked.

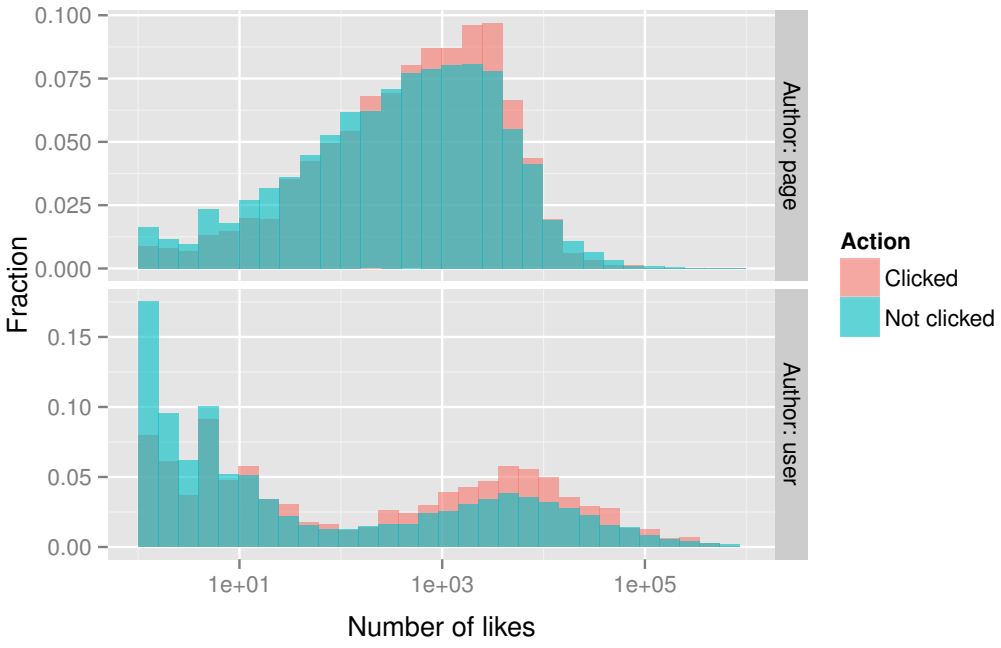


Figure 7.3: Distribution of clicked and unclicked videos with respect to the number of likes; stacked plots indicate discrete numbers

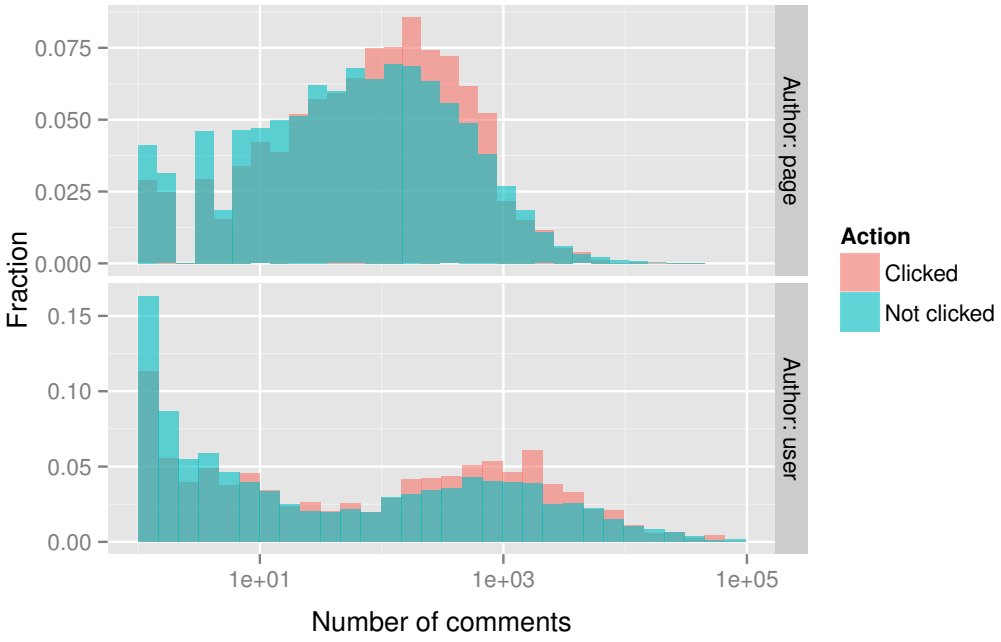


Figure 7.4: Distribution of clicked and unclicked videos with respect to the number of comments; stacked plots indicate discrete numbers

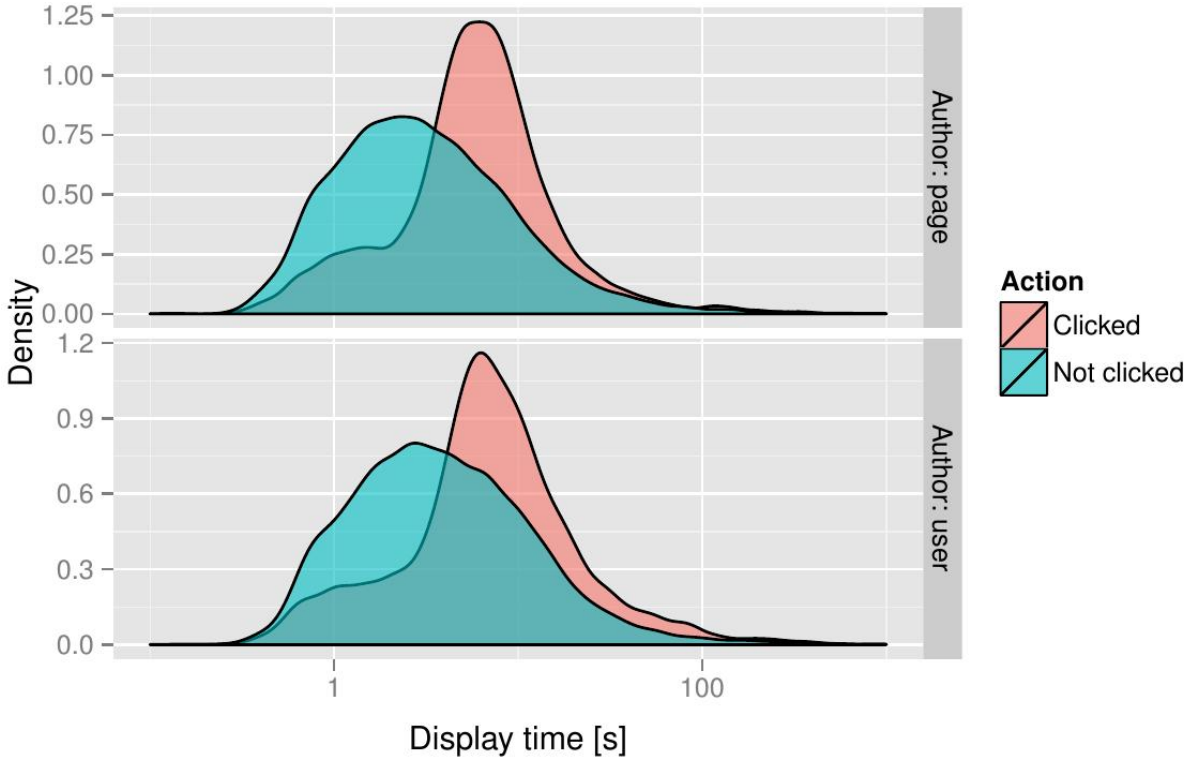


Figure 7.5: Duration of newsfeed entries to stay in the browser viewport either before being clicked or removed

7.3.2 Mobile Setting

Our prototypical evaluation on mobile devices shows that the results obtained in the large-scale, stationary setting can be mapped to mobile devices. Figure 7.7 illustrates the influence of likes and comments on videos. Both categories, the one of clicked videos and the one of non-clicked videos, are normalized to one. A large proportion of the videos is neither commented nor liked at all.

It highlights two phenomena observable in social networks: 1) Many of the video posts, clicked as well as non-clicked, have no likes or comments, and 2) a low number of likes and comments does not mean that the videos are not watched. It shows that many videos are consumed shortly after publishing, or that they are distributed to only small groups of friends. This result is supported by Table 7.2 which investigates friendlists.

The distance to a given user is approximated by her membership in a friendlist. We show in Table 7.2 only the subset of posts that has been shared by users within a friendlist. Despite the anonymization of traced data, standard Facebook groups such as 'family' and 'close friends' can still be identified. For both videos and photos, posts by close friends and family are preferred. The information on the social distance to a posting user can thus easily be identified and leveraged for prefetching mechanisms. The videos shared by close friends or family are predominantly those with low numbers of likes or comments. None of these videos in our dataset, reshared by close friends, has one thousand or more likes. It demonstrates that videos are either consumed

	Photos	Videos
Close Friends	70.1%	85.7%
Family	82.6%	50%
Other lists	9.2%	8.3%

Table 7.2: Friendlists and their impact on consuming video and photos. The table shows percentage of media shared by members of a friendlist that was clicked

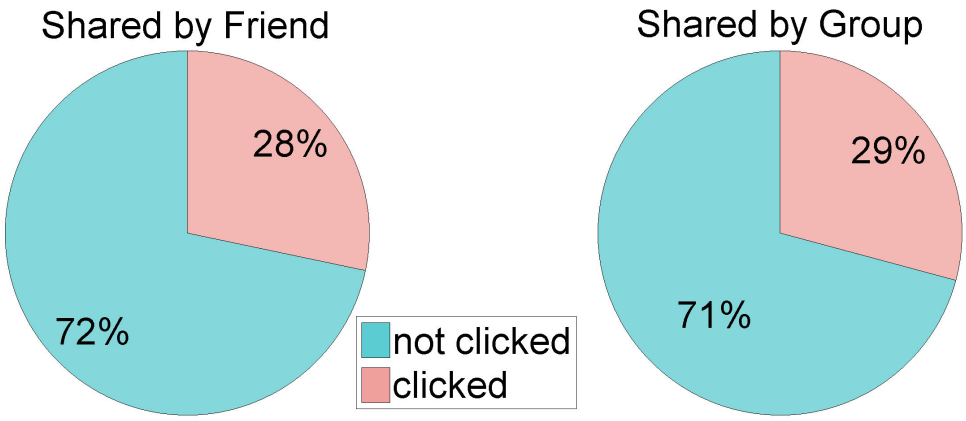


Figure 7.6: Impact of photos and videos shared by friends versus global Facebook groups or pages

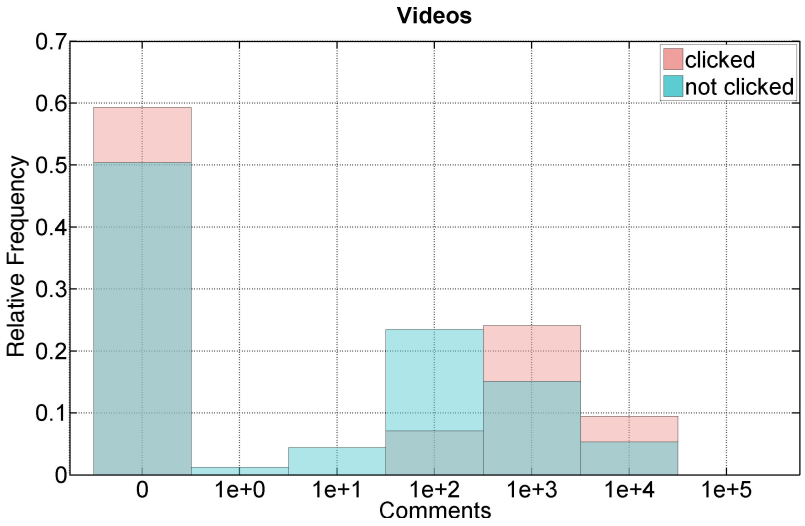
quickly after being shared with close friends or they don't have to be very popular in the social network to be watched by close friends of the videos' authors.

In total the proportion of unwatched videos is, as shown for stationary evaluation, nearly two times higher than those being watched. A significant outlier is located between 100 and 1000 likes. In this range, the percentage of clicked videos is nearly as high as for videos without likes.

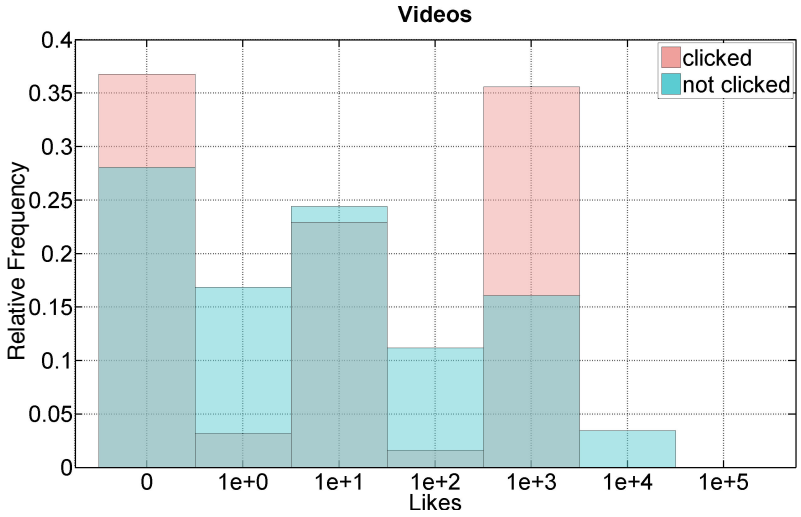
Comparing content shared by friends with content posted by a Facebook group, it can be seen that video consumption patterns are very similar. This is true for both videos and photos. Figure 7.6 illustrates that for both videos as well as photos the consumption behavior is the same as if the origin is a friend or a group.

7.4 Conclusion

In this chapter, a large-scale analysis on the impact of comments, likes and friends as originators on video consumption is presented. We discovered that short-time prefetching is an option to reduce the start-up delay of videos. The reason is that clicked posts remain longer in the browser viewport (before being clicked) than those that will not be clicked. This initial time can efficiently be leveraged to initiate download connections and stream the first chunks of a video. Since the exact time strongly depends on



(a) Effect of number of comments



(b) Effect of number of likes

Figure 7.7: Influence of the number of comments and likes on the consumption of videos on mobile devices

the user, we suggest to initiate download connections after two seconds as a rule of thumb.

Furthermore, we did not observe videos to have a high likelihood of being watched in case they count a high number of likes and comments. That indicates that prefetching mechanisms should not solely be based on the pure number of likes and comments. It is necessary to integrate measures such as closeness of a friend as originator.

We have also shown that being a friend with the video resharing person is no sufficient indicator for watching the latter. The type of friendship is important. 85.7% of the videos, shared by users that are part of the group 'close friends', and 50% of the videos from family members are watched. This is independent from the number of likes, comments and the freshness of the videos (in case that they are still displayed in the feed). Since only 8.3% of the other friends' videos are watched, we suggest to

prefetch the videos of authors which are labeled to be 'close friend' or family member. Relying on those both first major insights of this chapter, a next step is to design prefetching mechanisms for mobile devices that take the individual user characteristics into account.

Chapter 8

Summary, Conclusion and Future Work

In this chapter, we summarize the content of this thesis, draw and discuss conclusions from our work and depict future work in this research area. Unlike summaries of previous chapters on the individual contributions, this section provides a holistic view on the thesis.

8.1 Summary

The topic of this thesis is to avoid and mitigate adversarial side-effects of using OSNs. The spectrum of side-effects spans from having too many party guests[1], cyber bullying, facilitating social engineering attacks, undesired effects in job interviews (human resource managers may use OSNs to investigate the personality of applicants), losing the job due to improper public comments and even imprisoning in autocratic countries. Also being forced of granting content use licenses to OSN providers and censorship can be assumed not to be in line with OSN user's interests.

We identified two important causes for side-effects: First, sharing bits of information with too many recipients (e.g. the public) or with a wrong set of recipients (e.g. mistakenly chosen subset of friends). Second, most popular OSNs are run by omnipotent profit-oriented providers that monetize user data far beyond user's content sharing interests [Falch et al., 2009]. We argue that only a mixture of various user actions and novel approaches can effectively change this situation. In this thesis, we contributed to four main fields.

In the field of user behavior in OSNs, we contribute two large-scale user studies. In contrast to related work, we evaluate the behavior from real users who use Facebook on their own user profiles for their own reasons on their own devices.

User behavior in Facebook is changing at the scale of years. While low effort actions, such as likes and reshares, recently became more popular, the contribution of photos,

[1] http://www.stern.de/digital/online/facebook-fans-stuermen-geburtstagsparty-im-vorgarten-von-thessa-1692209.html, accessed on 2015-11-01

status updates and comments is relatively decreasing. Facebook also became mature and stable. This is not only indicated by a decreasing user growth rate but also by a decreasing establishment of new connections amongst the existing set of users. Users discover fewer new people to add as friend. Based on those observations, we argue that stale user behavior models may not reflect the recent situation in OSNs.

Sessions in Facebook are shorter and less frequent than assumed in the literature. In particular, very long sessions are missing. This is important for P2P-DOSNs since the session durations and session frequencies determine the churn model. The latter is a basic assumption on the amount of resources that are available in the network. Furthermore, shared content in Facebook is very fresh. 84.79% of all posts are not older than 24 hours until being viewed by the recipients. Only a negligible amount of content, accessed by Facebook users is older than one week. We thus advocate that alternative OSNs can be designed in a lightweight way, only storing the most recent data. This also allows to use the systems in environments with high churn and low resource contribution.

We also found that introducing the comprehensible color-coding interface of the FPW impacts the audience selection of users. However, the total amount of content which is visible to Facebook users does not dramatically decrease after introducing a comprehensible visualization of privacy controls, but the composition of the visible content changes. Which information is uploaded to Facebook as well as which information is shared with whom is strongly depending on the user's country of origin. Thus, global default privacy settings cannot meet the sharing interests of all users since the sharing interests show country-specific as well as person-specific differences.

In spite of the impact of C4PS on the visibility of content, users still need to trust the OSN provider not to misuse user's data and to protect it against external and internal (e.g. provider's employees) attackers. Since decentralizing OSNs abolishes both the explicit authority that needs to be trusted and ownership implications of OSNs being owned by commercial companies, we surveyed the state-of-the-art in the field of DOSNs. We found that DOSNs - except Diaspora - are academic approaches that only introduce concepts rather than being full-fledged OSNs. Also, there is a big gap in performance and functionality between today's OSNs and DOSN.

To that end, we proposed several improvements on DOSNs:

- a search scheme that allows to find user handles without disclosing information that is linked to a user handle,
- a lightweight storage concept for P2P-OSNs, applicable in very dynamic environments and
- a video prefetching approach to avoid delays in DOSNs, based on locally available information.

8.2 Conclusions

Privacy desires are diverse amongst users, strongly influenced by their country of origin. Thus, users need to be able to make qualified decisions and to be aware of possible consequences of user actions in OSNs. This should be learned and taught. Technical

solutions cannot take the responsibility of content publishers to choose their audience in OSNs without limiting the power of the communication system. The audience selection decisions can only be supported and simplified by technical means (e.g. privacy recommender systems, simple interfaces, audience views). Also, friend's information disclosures affect the own privacy because of inference attacks. Therefore, users should consider both sides: whether they want to befriend with others who are not concerned about privacy as well as whether friends are fine when publishing information that could harm their privacy.

Today's OSNs require users to trust the OSN providers not to misuse private data and to be able to protect the latter against adversaries. To the best of our knowledge, there is no feasible way to prevent OSN providers from learning valuable information about their users. Cryptography still allows OSN providers to evaluate ciphertext to infer many bits of information including user's social graph. Also, users of OSNs are subject of "engineering the public" [2] and legal issues such as the requirement to grant usage licenses on content, which is posted in OSNs, support advocates of the decentralization of OSNs.

The decentralized alternatives, DOSNs, suffer many drawbacks so far: They introduce new traffic observing opportunities (Section 4.8) and abolish the OSN provider's tool set to intervene in case of Cyber Bullying. Furthermore, DOSNs do not provide the same functionality and performance compared with their centralized counterparts. We address the functionality and performance issues in this thesis.

8.3 Future Work

In this section, we depict and discuss opportunities for future research. We identified future research opportunities in all fields that have been targeted in this thesis.

User Behavior in OSNs

Understanding user behavior means to investigate a moving target, quickly changing over time (Figure 2.19). OSN users undergo a learning curve in how to use the tool and hypes, trends, fads and even revolutionary upheavals (e.g. Arabic spring) impact OSN usage. Also, the technical environment changes (e.g. ubiquity of mobile devices), thus causing OSN users to adopt their behavior. In the literature, dynamics have been analyzed with respect to the development (life cycle) of online communities, the impact of incidents and trends and the evolution of personal interactions. So far, nobody either described the impact of these dynamics on OSN user models or integrated dynamics directly into the latter as a function over time. Future follow-up studies on the same subject seem to be useful to understand dynamics to integrate these dynamics into OSN user models.

We provide evidence to assume differences amongst users in different countries in Section 3.2.3. Those differences are important in case of building OSNs that address an international audience. However, there is no published comprehensive study that

[2] http://firstmonday.org/ojs/index.php/fm/article/view/4901/4097, accessed on 2015-11-01

covers all regions in the world (e.g. Asia vs. Africa) and compares usage patterns in detail.

Audience Selection

We demonstrated *C*4PS to be useful in Facebook and published the FPW to allow everybody to benefit from our work. Being a concept of general applicability, it makes sense to apply it on other social networking services in future work. Furthermore, since our color-coding approach is orthogonal to the related work, it can be combined with other approaches to further improve the audience selection in OSNs:

- Recommendation systems [Fang et al., 2010] for audience selection reduce the number of user action which are necessary to choose the desired audience. However, it does not totally abolish the need for manual adjustments. Users thus still need to be able to understand the manual privacy controls. C4PS can mitigate this burden.
- Venn diagrams [Egelman et al., 2011] are especially valuable tools when selecting various subsets of friends to be the audience.
- Our color-based interface allows to quickly grasp an overview of the actual privacy settings of the whole profile. However, the complexity that may occur in case of using various custom access rules for different content items is not mitigated by our approach. The concept of the audience view [Lipford et al., 2008] in combination with C4PS is useful to verify settings independent from their complexity.

Future work could develop strategies to combine these approaches, integrate them into a novel audience selection interface and measure the resulting usage errors and efforts.

DOSN

Many remaining challenges exist in the field of DOSNs. The state of the art DOSNs focus on basic features, such as user profile replication and private communication. In contrast, successful OSNs offer many features to make social networking a desirable experience. In comparison with OSNs, DOSNs suffer from lacks of performance and functionality. The scientific and the algorithmic challenges in bridging these lacks are to realize these features in a distributed and privacy preserving way, without relying on the global knowledge that an omnipotent OSN provider has.

Both, our search scheme as well as the prefetching mechanism are first steps towards enriching features of DOSNs and improving their performance. Future work in this field can be done by extending the search scheme to be applicable in P2P based and hybrid DOSNs (e.g. via super node based architecture) and by extending the prefetching by taking the history of personal content consumption habits into account. Additionally, integrating privacy preserving distributed recommender systems into DOSNs to provides useful content matchmaking features.

Appendices

Appendix A: Search Scheme Communication Cost Calculations

a	length of sender address
b	length of receiver address
H	length of the host name
K	length of key
R_i	length of the i-th field name
m	number of variable inner parts of the message
M	length of the message
n	number of variable parts
N	number of participating servers
Q	length of sequence number
S	length of the static part of the message
V_i	length of the i-th field value
X	length of UUID (= 36 as max. length)
Y	length of outer variable part

Register Phase

During registration (cmp. Sec. 5.3.2), four different message types are necessary to register a profile entry in the DHT.

1. *publicprofiledata/set* message (A): This message consists of one static part and one part per search field entry with a variable length, depending on the length of the field name and the length of the field value. The static part of message is defined as:

```
<iq from="a" to="b" type="set" id="X">
  <query xmlns="dhtsearch:publicprofiledata">
  </query>
</iq>
```

   The length S of static part is: $S = 91 + X + a + b$

   The variable part of message is:

```
<value fieldname="R_i">V_i</value>
```

The length Z of variable part with n subparts is:

$$Z = \sum_{i=0}^{n-1}(28 + R_i + V_i) \tag{.1}$$

guiding to the length M of the whole message with n variable subparts:

$$M = S + Z = 127 + a + b + 28n + \sum_{i=0}^{n-1}(R_i + V_i) \tag{.2}$$

2. *findsuccessor/get* message (B):

```
<iq from="a" to="b" type="get" id="X" seq="Q">
  <query xmlns="dhtsearch:findsuccessor">
    <key>K</key>
  </query>
</iq>
```

length M: $M = 106 + X + a + b + K + Q$

3. *findsuccessor/result* message (C):

```
<iq from="a" to="b"type="result" id="X"seq="Q">
  <query xmlns="dhtsearch:findsuccessor">
    <host>H</host>
  </query>
</iq>
```

length M: $M = 111 + X + a + b + H + Q$

4. *storekey/set* message (D):

```
<iq from="a" to="b"type="set" id="X"seq="Q">
  <query xmlns="dhtsearch:findsuccessor">
    <host>H</host>
  </query>
</iq>
```

length M: $M = 111 + X + a + b + H + Q$

Depending on the number of fields (n) in the profile, the number of messages, which is needed on the registration phase, varies. The effort of registration is:

- 1 message of type A with n entries
- $n \cdot log(N)$ messages of type B
- $n \cdot log(N)$ messages of type C
- n messages of type D

The total *Data Volume (V)* which is necessary to register a profile at the lookup service can be calculated by:

$$V = |A| + n \cdot log(N) \cdot |B| + n \cdot log(N) \cdot |C| + n \cdot |D|$$

Search Phase

Conducting a search procedure (Fig. 5.1) needs 8 different types of messages. In this section, we provide formulas to calculate their size and the overall traffic, caused by this action. Again, some messages have of request-depending variable parts.

1. *searchservers/get* message (*A*) static part of message:

```
<iq from="a" to="b" type="get" id="X">
  <query xmlns="dhtsearch:searchservers">
  </query>
</iq>
```

   length S of static part:

$$S = 88 + X + a + b = 124 + a + b \tag{.3}$$

   variable part of message:

```
<value fieldname="Ri">Vi</value>
```

   length Z of variable part with n subparts:

$$Z = \sum_{i=0}^{n-1} (28 + R_i + V_i) \tag{.4}$$

   length M of whole message with n subparts in the variable part:

$$M = S + Z = 124 + a + b + 28n + \sum_{i=0}^{n-1} (R_i + V_i) \tag{.5}$$

2. *findsuccessor/get* message (*B*)

   Cmp. register phase (2)

3. *findsuccessor/result* message (*C*)

   Cmp. register phase (3)

4. *searchhosts/get* message (*D*)

```
<iq from="a" to="b" type="get" id="X">
  <query xmlns="dhtsearch:searchhosts">
  <key>K</key>
  </query>
</iq>
```

   length M of static part:

$$M = 96 + X + a + b + K = 132 + a + b + K \tag{.6}$$

5. *searchhosts/result* message (*E*)

   static part of message:

```
<iq from="a" to="b" type="result" id="X">
  <query xmlns="dhtsearch:searchhosts">
  </query>
</iq>
```

length S of static part:

$$S = 88 + X + a + b = 124 + a + b \tag{.7}$$

variable part of message:

```
<host>H_i</host>
```

length Z of variable part with n subparts:

$$Z = \sum_{i=0}^{n-1} (13 + H_i) \tag{.8}$$

length M of whole message with n subparts in the variable part:

$$M = S + Z = 124 + a + b + 13n + \sum_{i=0}^{n-1} (H_i) \tag{.9}$$

6. *searchservers/result* message (F)

   static part of message:

```
<iq from="a" to="b" type="␣result␣" id="X">
<query xmlns="dhtsearch:searchservers">
</query>
</iq>
```

   length S of static part:

$$S = 92 + X + a + b = 128 + a + b \tag{.10}$$

   variable part of message:

```
<host>H_i</host>
```

   length Z of variable part with n subparts:

$$Z = \sum_{i=0}^{n-1} (13 + H_i) \tag{.11}$$

   length M of whole message with n subparts in the variable part:

$$M = S + Z = 128 + a + b + 13n + \sum_{i=0}^{n-1} (H_i) \tag{.12}$$

7. *searchonserver/get* message (G)

   static part of message:

```
<iq from="a" to="b" type="get" id="X">
 <query xmlns="dhtsearch:searchonserver">
 </query>
</iq>
```

   length M: $M = 88 + X + a + b = 124 + a + b$

   variable part of message:

```
<value fieldname="R_i">V_i</value>
```

   length Z of variable part with n subparts:

$$Z = \sum_{i=0}^{n-1} (28 + R_i + V_i) \tag{.13}$$

   length M of whole message with n subparts in the variable part:

$$M = S + Z = 124 + a + b + 28n + \sum_{i=0}^{n-1} (R_i + V_i) \tag{.14}$$

8. *serchonserver/result* message (H)

   static part of message:

```
<iq from="a" to="b" type="result" id="X">
  <query xmlns="dhtsearch:searchonserver">
  </query>
</iq>
```

   length S of static part: $S = 91 + X + a + b = 127 + a + b$

   variable outer part of message:

```
<user searchid="X"></user>
```

   length Y of variable outer part with n subparts: $Y = 25 + X = 61$

   variable inner part of message:

```
<value fieldname="c_m">d_m</value>
```

   length Z of variable part with m subparts: $Z = \sum_{i=0}^{m-1} (28 + R_i + V_i)$

   length M of the whole message with m inner subparts in the n outer variable parts: $M = S + n \cdot Y + n \cdot Z_{m_n}$

Data volume for finding IDs(V):

- 1 message of type A with j variable parts x
- $j \cdot log(N)$ messages of type B
- $j \cdot log(N)$ messages of type C
- j messages of type D
- j messages of type E with k variable parts per message
- l messages of type F with l variable parts per message
- 2 messages of type G with m variable parts per message
- 2 message of type H with n variable parts for users and p variable parts for profile data

This formula describes the amount of data, caused by one search request:

$$V = |A(j)| + j \cdot log(N) \cdot |B| + j \cdot log(N) \cdot |C| + j \cdot |D| \\ + j \cdot |E(k)| + |F(l)| + 2 \cdot |G(2^{m-2})| + 2 \cdot |H(n,p)| \tag{.15}$$

Appendix B: C4PS Questionnaire

Proband Lfd. Nr.: _____ Datum: _____

[Vor dem praktischen Teil:]

1.) Wie häufig nutzen Sie Online-Social-Networks?

❑ Mehrmals täglich ❑ Täglich ❑ Mehrmals pro Woche ❑ Seltener ❑ Nie

2.) Benutzen Sie Facebook? ❑ Ja ❑ Nein

3.) Welche anderen Online-Social-Networks benutzen Sie?

4.) Haben Sie sich bisher mit den Privatsphäre-Einstellungen der Plattformen auseinandergesetzt? ❑ Ja ❑ Nein

5.) Als wie einfach/übersichtlich empfanden Sie diese Einstellungen?

❑ Sehr übersichtlich ❑ übersichtlich ❑ unübersichtlich ❑ Sehr unübersichtlich

6.) Wie oft verändern/überprüfen Sie diese Einstellungen?

❑ Wöchentlich ❑ Monatlich ❑ Seltener ❑ Gar nicht

7.) Benutzen Sie die Möglichkeit Gruppen / Listen von Freunden anzulegen? ❑ Ja ❑ Nein
(z.B. „Schulfreunde“, „Arbeitskollegen“, ...)

Wenn „Nein“: **Warum nicht?**

8.) Wie oft verändern Sie diese Gruppen? (Nutzer hinzufügen bzw. löschen)

❑ Wöchentlich ❑ Monatlich ❑ Seltener ❑ Gar nicht

9.) Nutzen Sie die Möglichkeit, spezielle Rechte für einzelne Benutzer/Gruppen einzustellen? ❑ Ja ❑ Nein

10.) Sind Sie sich bewusst, welche Informationen genau für andere Personen sichtbar sind? ❑ Ja ❑ Nein

11.) Was denken Sie, wie gut ihre derzeitigen Privatsphäre-Einstellungen das Vertrauen in andere Nutzer widerspiegeln?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

Proband Lfd. Nr.: _____ Datum: _____

[Praktischer Teil – neues System]

1.) Für welche Benutzer/Gruppen ist das Attribut „Geburtstag“ sichtbar?

2.) Für welche Benutzer/Gruppen ist das Attribut „Derzeitiger Wohnort/Heimatstadt“ sichtbar?

3.) Für welche Benutzer/Gruppen ist das Attribut „Beziehungsstatus“ sichtbar?

4.) Für welche Benutzer/Gruppen ist das Fotoalbum „Junggesellenabschied“ sichtbar?

5.) Welche Attribute des Profils sind für den Benutzer „Stephanie Schmidt“ sichtbar?

6.) Legen Sie eine neue Gruppe mit dem Namen „gute Freunde“ an

7.) Fügen Sie in die neu erstelle Gruppe „gute Freunde“ alle Mitglieder aus der Gruppe „Schulfreunde“ ein und zusätzlich noch die beiden Freunde „Claudia Bauer“ und „Sun Yen“.

- bitte wenden -

Proband Lfd. Nr.: _____ Datum: _____

8.) Stellen Sie die Privatsphäre ihres Profils so ein, dass die Felder wie folgt sichtbar sind:

- Handy-Nummer:	Nur „Jan Weber“ und „Daniela Faber“
- „Gefällt mir“ und Interessen:	Jeder
- Derzeitiger Wohnort / Heimatort:	nur die Gruppe „Schulfreunde“
- Beziehungsstatus:	Niemand
- Religiöse Ansichten / politische Einstellung:	Alle Freunde

9.) Stellen Sie die Privatsphäre des Fotoalbums „Junggesellenabschied“ so ein, dass es von der Gruppe „gute Freunde“, allerdings nicht von „Patrick Maur“ gesehen werden kann.

10.) Beantworten Sie die folgenden Fragen, in dem Sie das entsprechende Kästchen ankreuzen (✗). Ganz links bedeutet „stimme überhaupt nicht zu“ und ganz rechts bedeutet „ich stimme voll und ganz zu“. Die Kästchen dazwischen dienen zur Abstufung.

		Stimme überhaupt nicht zu				Stimme voll und ganz zu
1.	Ich würde das System gerne öfter benutzen.	❑	❑	❑	❑	❑
2.	Ich fand das System unnötig kompliziert.	❑	❑	❑	❑	❑
3.	Das System war einfach zu benutzen.	❑	❑	❑	❑	❑
4.	Ich denke, dass ich Unterstützung bräuchte, um das System zu benutzen.	❑	❑	❑	❑	❑
5.	Die verschiedenen Funktionen waren gut in das System integriert.	❑	❑	❑	❑	❑
6.	Es waren zu viele Unstimmigkeiten im System vorhanden.	❑	❑	❑	❑	❑
7.	Ich denke, die meisten Leute würden den Umgang mit dem System schnell erlernen.	❑	❑	❑	❑	❑
8.	Das System ließ sich sehr umständlich benutzen.	❑	❑	❑	❑	❑
9.	Ich war sehr sicher im Umgang mit dem System	❑	❑	❑	❑	❑
10.	Ich musste eine Menge Dinge lernen, bevor ich mit diesem System loslegen konnte.	❑	❑	❑	❑	❑

Proband Lfd. Nr.: _____ Datum: _____

[Praktischer Teil – Facebook]

1.) Für welche Benutzer/Liste ist das Attribut „Geburtstag“ sichtbar?

2.) Für welche Benutzer/Liste ist das Attribut „Derzeitiger Wohnort/Heimatstadt“ sichtbar?

3.) Für welche Benutzer/Liste ist das Attribut „Beziehungen“ sichtbar?

4.) Für welche Benutzer/Liste ist das Fotoalbum „Junggesellenabschied“ sichtbar?

5.) Welche Attribute des Profils sind für den Benutzer „Stephanie Schmidt“ sichtbar?

6.) Legen Sie eine neue Liste mit dem Namen „gute Freunde“ an

7.) Fügen Sie in die neu erstelle Liste „gute Freunde“ alle Mitglieder aus der Liste „Schulfreunde“ ein und zusätzlich noch die beiden Freunde „Claudia Bauer“ und „Sun Yen“.

- bitte wenden -

Proband Lfd. Nr.: _____ Datum: _____

8.) Stellen Sie die Privatsphäre ihres Profils so ein, dass die Felder wie folgt sichtbar sind:

- Handy-Nummer:	Nur „Jan Weber“ und „Daniela Faber“
- Interessen und Seiten:	Jeder
- derzeitiger Wohnort / Heimatstadt:	nur die Liste „Schulfreunde“
- Beziehungen:	Niemand
- Religiöse Ansichten / politische Einstellung:	Nur Freunde

9.) Stellen Sie die Privatsphäre des Fotoalbums „Junggesellenabschied“ so ein, dass es von der Liste „Schulfreunde“, allerdings nicht von „Patrick Maur“ gesehen werden kann.

Proband Lfd. Nr.: _____ Datum: _____

[Nach dem praktischen Teil]

1.) Wie beurteilen Sie allgemein das bestehende System für Privatsphäre-Einstellungen bei Facebook?

❑ sehr übersichtlich ❑ übersichtlich ❑ etwas unübersichtlich ❑ unübersichtlich

2.) Wie finden Sie das neue System für Privatsphäre-Einstellungen im Vergleich zum bestehenden System bei Facebook?

❑ viel besser ❑ etwas besser ❑ etwas schlechter ❑ viel schlechter

3.) Wie gut gefällt Ihnen die Darstellung der Privatsphäre über einzelne Farben?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

4.) Sind die gewählten Farben für Sie sinnvoll/eindeutig? ❑ Ja ❑ Nein

5.) Wie gefällt Ihnen die Profil-Vorschau bei Facebook?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

6.) Wie gefällt Ihnen die Profil-Vorschau beim neuen System?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

7.) Wie kommen Sie mit dem Gruppen-Management bei Facebook zurecht?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

8.) Wie kommen Sie mit dem Gruppen-Management bei dem neuen System zurecht?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

9.) Wie einfach war für Sie das Einstellen von Sichtbarkeiten für einzelne Freunde/Gruppen bei Facebook?

❑ Sehr einfach ❑ einfach ❑ weniger einfach ❑ schwierig

10.) Wie einfach war für Sie das Einstellen von Sichtbarkeiten für einzelne Freunde/Gruppen bei dem neuen System?

❑ Sehr einfach ❑ einfach ❑ weniger einfach ❑ schwierig

11.) Was hat Ihnen an dem neuen System besonders gefallen?

- bitte wenden -

Proband Lfd. Nr.: _____ Datum: _____

12.) Was kann man an dem neuen System noch verbessern?

13.) Werden Sie in Zukunft Ihre Privatsphäre-Einstellungen in Online-Social-Networks verändern bzw. öfter kontrollieren? ❑ Ja ❑ Nein

14.) Was denken Sie, wie gut ihre derzeitigen Privatsphäre-Einstellungen das Vertrauen in andere Nutzer widerspiegeln?

❑ Sehr gut ❑ gut ❑ weniger gut ❑ schlecht

15.) Hat sich Ihr Bewusstsein über Ihre Privatsphäre im Internet nach dieser Studie verändert? ❑ Ja ❑ Nein

Platz für weitere Kommentare:

Demographische Angaben:

Geschlecht: ❑ männlich ❑ weiblich

Alter: _____

Berufs- / Studienrichtung: _____

Appendix C: List of Publications

Journal Publications

[Paul et al., 2014a] Thomas Paul and Antonino Famulari and Thorsten Strufe. (2014). A survey on decentralized Online Social Networks. In: Computer Networks.

[Gebelein et al., 2015b] Paul Gebelein, Thomas Paul, Thorsten Strufe, Wolfgang Effelsberg. (2015). Interdisziplinäre Forschung zwischen Informatik und Soziologie. In: PIK - Praxis der Informationsverarbeitung und Kommunikation.

Conference Publications

[Paul et al., 2012a] Thomas Paul, Martin Stopczynski, Daniel Puscher, Melanie Volkamer, Thorsten Strufe. (2012). C4PS - Colors for Privacy Settings. In: WWW 2012.

[Paul et al., 2014b] Thomas Paul, Marius Hornung, Thorsten Strufe. (2014). Distributed Discovery of User Handles with Privacy. GlobeCom 2014.

[Paul et al., 2012b] Thomas Paul and Martin Stopczynski and Daniel Puscher and Melanie Volkamer and Thorsten Strufe. (2012). C4PS - Helping Facebookers Manage their Privacy Settings SocInfo 2012.

[Paul et al., 2015c] Thomas Paul, Daniel Puscher, Stefan Wilk, Thorsten Strufe. (2015). Systematic, Large-scale Analysis on the Feasibility of Media Prefetching in Online Social Networks. CCNC, 2015.

[Paul et al., 2015d] Thomas Paul, Stephen Stephen, Hani Salah, Thorsten Strufe. (2015). The Students' Portal of Ilmenau: A Holistic OSN's User Behaviour Model. PICCIT 2015

Book Chapters

[Paul et al., 2011a] Thomas Paul and Sonja Buchegger and Thorsten Strufe. (2011). Handbook on the Trustworthy Internet. In: Giuseppe Bianchi, chap. Decentralizing Social Networking Services, Springer.

[Gebelein et al., 2015a] Paul Gebelein, Martina Löw, Thomas Paul. (2015). Flash Mobs als Innovation. Über eine neue Sozialform technisch vermittelter Versammlung. In: Rammert, Werner (Hrsg.): Innovationsgesellschaft heute.

Other Publications

[Paul et al., 2010a] Thomas Paul, Sonja Buchegger, Thorsten Strufe. (2010). Decentralizing Social Networking Services. International Tyrrhenian Workshop on Digital Communications.

[Paul et al., 2011b] Thomas Paul and Thorsten Strufe. (2011). Improving the Usability of Privacy Settings in Facebook. In: Proceedings of HCI/Health, Wealth and Identity Theft.

[Paul et al., 2011c] Thomas Paul, Daniel Puscher, and Thorsten Strufe. (2011). Improving the Usability of Privacy Settings in Facebook no. arXiv:1109.6046v1.

[Paul et al., 2012c] Paul, T., Greschbach, B., Buchegger, S., and Strufe, T. (2012a). Exploring decentralization dimensions of social network services: Adversaries and availability. In HotSoc (KDD Workshop).

[Paul et al., 2015a] Thomas Paul, Daniel Puscher, and Thorsten Strufe. (2015). Private Date Exposure in Facebook and the Impact of Comprehensible Audience Selection Controls.arXiv preprint arXiv:1505.06178.

[Paul et al., 2015b] Thomas Paul, Daniel Puscher, and Thorsten Strufe. (2015). The User Behavior in Facebook and its Development from 2009 until 2014. arXiv preprint arXiv:1505.04943.

Bibliography

[Aiello et al., 2008] Aiello, L. M., Milanesio, M., Ruffo, G., and Schifanella, R. (2008). Tempering Kademlia with a robust identity based system. In *IEEE P2P*.

[Aiello and Ruffo, 2012] Aiello, L. M. and Ruffo, G. (2012). Lotusnet: Tunable privacy for distributed online social network services. *Computer Communications*, 35.

[Anderson et al., 2009] Anderson, J., Bonneau, J., Diaz, C., and Stajano, F. (2009). Privacy-enabling social networking over untrusted networks. *WOSN*.

[Armknecht et al., 2014] Armknecht, F. et al. (2014). Protecting Public OSN Posts from Unintended Access. In *ICC*.

[Backstrom et al., 2011] Backstrom, L., Bakshy, E., Kleinberg, J. M., Lento, T. M., and Rosenn, I. (2011). Center of attention: How facebook users allocate attention across friends. *ICWSM*.

[Baden et al., 2009] Baden, R., Bender, A., Spring, N., Bhattacharjee, B., and Starin, D. (2009). Persona: An online social network with user-defined privacy. *SIGCOMM*.

[Bai et al., 2013] Bai, X., Junqueira, F. P., and Silberstein, A. (2013). Cache refreshing for online social news feeds. In *CIKM*. ACM.

[Bangor et al., 2009] Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*.

[Baumgart et al., 2007] Baumgart, I., Heep, B., and Krause, S. (2007). OverSim: A Flexible Overlay Network Simulation Framework. *IEEE Global Internet Symposium*.

[Benevenuto et al., 2009] Benevenuto, F., Rodrigues, T., Cha, M., and Almeida, V. (2009). Characterizing user behavior in online social networks. In *IMC*.

[Benjamini and Hochberg, 1995] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*.

[Benvenuto et al., 2009] Benvenuto, F., Rordrigues, T., Cha, M., and Almeida, V. (2009). Characterizing user behavior in online social network. *IMC*.

[Bessani et al., 2011] Bessani, A., Correia, M., Quaresma, B., André, F., and Sousa, P. (2011). Depsky: dependable and secure storage in a cloud-of-clouds. In *Sixth conference on Computer systems*. ACM.

[Bodriagov and Buchegger, 2012] Bodriagov, O. and Buchegger, S. (2012). P2p social networks with broadcast encryption protected privacy. In *Privacy and Identity Management for Life*. Springer.

[Boneh et al., 2005] Boneh, D., Boyen, X., and Goh, E.-J. (2005). Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology – EUROCRYPT 2005*, LNCS. Springer.

[Boneh et al., 2004] Boneh, D. et al. (2004). Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt*.

[Bonifati et al., 2004] Bonifati, A. et al. (2004). Xpath lookup queries in p2p networks. In *WIDM*.

[Breslau et al., 1999] Breslau, L. et al. (1999). Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM*.

[Brooke, 1996] Brooke, J. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*.

[Buchegger et al., 2009a] Buchegger, S. et al. (2009a). PeerSoN: P2P Social Networking - Early Experiences and Insights. In *SNS*.

[Buchegger et al., 2009b] Buchegger, S., Schioberg, D., Vu, L., and Datta, A. (2009b). Peerson: P2p social networking - early experiences and insights. *SNS*.

[Campbell, 2005] Campbell, M. A. (2005). Cyber bullying: An old problem in a new guise?. *Australian journal of Guidance and Counselling*.

[Carminati et al., 2009] Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. (2009). A semantic web based framework for social network access control. In *SACMAT*.

[Carminati et al., 2006] Carminati, B., Ferrari, E., and Perego, A. (2006). Rule-based access control for social networks. *On the Move to Meaningful Internet . . . .*

[Catanese et al., 2011] Catanese, S. A., De Meo, P., Ferrara, E., Fiumara, G., and Provetti, A. (2011). Crawling facebook for social network analysis purposes. In *WIMS*.

[Cha et al., 2007] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. (2007). I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. In *IMC*.

[Cheng et al., 2009] Cheng, X., Fraser, S., and Liu, J. (2009). Nettube: Exploring social networks for peer-to-peer short video sharing. In *IEEE INFOCOM*.

[Cutillo et al., 2009a] Cutillo, A., Molva, R., and Strufe, T. (2009a). Safebook: a privacy preserving online social network leveraging on real-life trust. *IEEE Communication Magazine*.

[Cutillo et al., 2009b] Cutillo, A., Molva, R., and Strufe, T. (2009b). Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network. In *WoWMoM*.

[Cutillo et al., 2009c] Cutillo, L.-A., Molva, R., and Strufe, T. (2009c). Safebook: a privacy preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*.

[De Cristofaro et al., 2013] De Cristofaro, E., Manulis, M., and Poettering, B. (2013). Private discovery of common social contacts. *International Journal of Information Security*.

[Delerablée, 2007] Delerablée, C. (2007). Identity-based broadcast encryption with constant size ciphertexts and private keys. In *Advances in Cryptology – ASIACRYPT*.

[Druschel and Rowstron, 2001] Druschel, P. and Rowstron, A. (2001). Past: A large-scale, persistent peer-to-peer storage utility. In *Hot Topics in Operating Systems*.

[Durr et al., 2012] Durr, M., Maier, M., and Dorfmeister, F. (2012). Vegas – a secure and privacy-preserving peer-to-peer online social network. In *Privacy, Security, Risk and Trust (PASSAT)*.

[Dwyer et al., 2007] Dwyer, C., Hiltz, S. R., and Passerini, K. (2007). Trust and privacy concern within social networking sites: A comparison of facebook and myspace. In *AMCIS*.

[Egelman et al., 2011] Egelman, S., Oates, A., and Krishnamurthi, S. (2011). Oops, i did it again: mitigating repeated access control errors on facebook. CHI '11.

[Falch et al., 2009] Falch, M., Henten, A., Tadayoni, R., and Windekilde, I. (2009). Business models in social networking. *CMI International Conference - Social Networking and Communities*.

[Famulari and Hecker, 2012] Famulari, A. and Hecker, A. (2012). Mantle: a novel dosn leveraging free storage and local software. In *ICAIT*.

[Fang et al., 2010] Fang, L., Kim, H., LeFevre, K., and Tami, A. (2010). A Privacy Recommendation Wizard for Users of Social Networking Sites. In *CCS*.

[Gautam et al., 2013] Gautam, N., Petander, H., and Noel, J. (2013). A comparison of the cost and energy efficiency of prefetching and streaming of mobile video. In *MoVid*. ACM Press.

[Goyal et al., 2006] Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*.

[Graffi et al., 2008] Graffi, K., Podrajanski, S., Mukherjee, P., Kovacevic, A., and Sreinmetz, R. (2008). A dsitributed platform for multimedia communities. *International Symposium on Multimedia*.

[Greschbach and Buchegger, 2012] Greschbach, B. and Buchegger, S. (2012). Friendly surveillance – a new adversary model for privacy in decentralized online social networks. In *Current Issues in IT Security*.

[Greschbach et al., 2012] Greschbach, B., Kreitz, G., and Buchegger, S. (2012). The devil is in the metadata - new privacy challenges in decentralised online social networks. In *SESOC (PERCOM Workshops)*.

[Gross and Acquisti, 2005] Gross, R. and Acquisti, A. (2005). Information revelation and privacy in online social networks. WPES.

[Guha et al., 2006] Guha, S., Daswani, N., and Jain, R. (2006). An Experimental Study of the Skype Peer-to-Peer VoIP System. In *IPTPS*.

[Guha et al., 2008] Guha, S., Tang, K., and Francis, P. (2008). NOYB: Privacy in Online Social Networks. In *WOSP*.

[Gummadi et al., 2003] Gummadi, K. P. et al. (2003). Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP*.

[Günther et al., 2012] Günther, F., Manulis, M., and Strufe, T. (2012). Cryptographic treatment of private user profiles. In *Financial Cryptography and Data Security*. Springer.

[Gyarmati and Trinh, 2010] Gyarmati, L. and Trinh, T. (2010). Measuring user behavior in online social networks. *Network, IEEE*.

[Huber et al., 2009] Huber, M., Kowalski, S., Nohlberg, M., and Tjoa, S. (2009). Towards automating social engineering using social networking sites. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*.

[Jahid et al., 2011] Jahid, S., Mittal, P., and Borisov, N. (2011). EASiER: encryption-based access control in social networks with efficient revocation. In *ASIACCS*.

[Jahid et al., 2012] Jahid, S., Nilizadeh, S., Mittal, P., Borisov, N., and Kapadia, A. (2012). Decent: A decentralized architecture for enforcing privacy in online social networks. *SESOC (PERCOM Workshops)*, pages 326–332.

[Jiang et al., 2013] Jiang, J., Wilson, C., Wang, X., Sha, W., Huang, P., Dai, Y., and Zhao, B. Y. (2013). Understanding latent interactions in online social networks. *ACM Transactions on the Web (TWEB)*.

[John Rose and Christine Barton and Robert Souza and James Platt, 2014] John Rose and Christine Barton and Robert Souza and James Platt (2014). Data Privacy by the Numbers. `https://www.bcgperspectives.com/content/Slideshow/information_technology_strategy_digital_economy_data_privacy_by_the_numbers/#ad-image-3`. accessed: 2014-08-6.

[Johnson et al., 2012] Johnson, M., Egelman, S., and Bellovin, S. M. (2012). Facebook and privacy: it's complicated. In *SOUPS*.

[Kaafar et al., 2013] Kaafar, M. A., Berkovsky, S., and Donnet, B. (2013). On the potential of recommendation technologies for efficient content delivery networks. *SIGCOMM Comput. Commun. Rev.*

[Karnstedt et al., 2007] Karnstedt, M., Sattler, K.-U., Richtarsky, M., Muller, J., Hauswirth, M., Schmidt, R., and John, R. (2007). Unistore: querying a dht-based universal storage. In *ICDE*.

[Khemmarat et al., 2011] Khemmarat, S., Zhou, R., Krishnappa, D., and Gao, L. (2011). Watching user generated videos with prefetching. In *MMSys*.

[Koll et al., 2013] Koll, D., Li, J., and Fu, X. (2013). With a little help from my friends: replica placement in decentralized online social networks. Technical report, Technical Report IFI-TB-2013-01, Institute of Computer Science, University of Goettingen, Germany.

[Krishnamurthy and Wills, 2008] Krishnamurthy, B. and Wills, C. E. (2008). Characterizing privacy in online social networks. In *WOSN*.

[Krishnan and Sitaraman, 2012] Krishnan, S. S. and Sitaraman, R. K. (2012). Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *IMC*, pages 211–224. ACM.

[Krug, 2005] Krug, S. (2005). *Don't Make Me Think: A Common Sense Approach to the Web (2nd Edition)*. New Riders Publishing.

[Kryczka et al., 2010] Kryczka, M., Cuevas, R., and Guerrero, C. (2010). A first step towards user assisted online social networks. *SNS*.

[Kubiatowicz et al., 2000] Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., et al. (2000). Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*.

[Leng, 2012] Leng, C. (2012). *BubbleStorm: Replication, Updates, and Consistency in Rendezvous Information Systems*. PhD thesis, Technische Universität Darmstadt.

[Li et al., 2013] Li, H., Wang, H., Liu, J., and Xu, K. (2013). Video requests from online social networks: Characterization, analysis and generation. In *IEEE INFOCOM*.

[Li et al., 2012] Li, Z., Shen, H., Wang, H., Liu, G., and Li, J. (2012). Socialtube: P2p-assisted video sharing in online social networks. In *INFOCOM*.

[Lin et al., 2012] Lin, J., Li, Z., Wang, D., Salamatian, K., and Xie, G. (2012). Analysis and comparison of interaction patterns in online social network and social media. In *ICCCN*.

[Lindamood et al., 2009] Lindamood, J. et al. (2009). Inferring Private Information Using Social Network Data. In *WWW*.

[Ling and Datta, 2014] Ling, C. and Datta, A. (2014). Intercloud raider: A do-it-yourself multi-cloud private data backup system. In *ICDCN*.

[Lipford et al., 2008] Lipford, H. R., Besmer, A., and Watson, J. (2008). Understanding Privacy Settings in Facebook with an Audience View. In *UPSEC*.

[Liu et al., 2011a] Liu, D., Shakimov, A., Càceres, R., Varshavsky, A., and Cox, L. P. (2011a). Confidant: Protecting osn data without locking it up. In *Middleware*.

[Liu et al., 2011b] Liu, Y., Gummadi, K. P., Krishnamurthy, B., and Mislove, A. (2011b). Analyzing facebook privacy settings: user expectations vs. reality. In *IMC*.

[Lopes and Ferreira, 2010] Lopes, P. and Ferreira, R. A. (2010). Splitquest: controlled and exhaustive search in peer-to-peer networks. In *IPTPS*.

[Luarn et al., 2014] Luarn, P., Yang, J.-C., and Chiu, Y.-P. (2014). The network effect on information dissemination on social network sites. *Computers in Human Behavior*.

[Madejski et al., 2011] Madejski, M., Johnson, M., and Bellovin, S. (2011). The Failure of Online Social Network Privacy Settings. Technical report, Columbia University.

[Mazzia et al., 2012] Mazzia, A., LeFevre, K., and Adar, E. (2012). The pviz comprehension tool for social network privacy settings. In *SOUPS*.

[McDonald and Thompson, 2015] McDonald, P. and Thompson, P. (2015). Social media(tion) and the reshaping of public/private boundaries in employment relations. *International Journal of Management Reviews*.

[Mega et al., 2011] Mega, G., Montresor, A., and Picco, G. P. (2011). Efficient dissemination in decentralized social networks. In *IEEE P2P*.

[Meo et al., 2014] Meo, P. d., Ferrara, E., Abel, F., Aroyo, L., and Houben, G.-J. (2014). Analyzing user behavior across social sharing environments. *ACM Trans. Intell. Syst. Technol.*

[Mondal et al., 2014] Mondal, M., Liu, Y., Viswanath, B., Gummadi, K. P., and Mislove, A. (2014). Understanding and specifying social access control lists. In *SOUPS*.

[Narayanan et al., 2012] Narayanan, A., Toubiana, V., Barocas, S., Nissenbaum, H., and Boneh, D. (2012). A critical look at decentralized personal data architectures. *arXiv preprint arXiv:1202.4503*.

[Narendula et al., 2012] Narendula, R., Papaioannou, T. G., and Aberer, K. (2012). A decentralized online social network with efficient user-driven replication. In *PASSAT*. IEEE.

[Nilizadeh et al., 2012] Nilizadeh, S., Jahid, S., Mittal, P., Borisov, N., and Kapadia, A. (2012). Cachet: A decentralized architecture for privacy preserving social networking with caching. In *CoNEXT*.

[Paul et al., 2014a] Paul, T., Famulari, A., and Strufe, T. (2014a). A survey on decentralized online social networks. *Computer Networks*.

[Paul et al., 2012a] Paul, T., Greschbach, B., Buchegger, S., and Strufe, T. (2012a). Exploring decentralization dimensions of social network services: Adversaries and availability. In *HotSoc (KDD Workshop)*.

[Paul et al., 2012b] Paul, T., Greschbach, B., Buchegger, S., and Strufe, T. (2012b). Exploring decentralization dimensions of social networking services: adversaries and availability. In *HotSoc (KDD Workshop)*.

[Paul et al., 2014b] Paul, T., Hornung, M., and Strufe, T. (2014b). Distributed discovery of user handles with privacy. In *Global Communications Conference (GLOBECOM), 2014 IEEE*.

[Paul et al., 2015a] Paul, T., Puscher, D., and Strufe, T. (2015a). Private date exposure in facebook and the impact of comprehensible audience selection controls. *arXiv preprint arXiv:1505.06178*.

[Paul et al., 2015b] Paul, T., Puscher, D., and Strufe, T. (2015b). The user behavior in facebook and its development from 2009 until 2014. *arXiv preprint arXiv:1505.04943*.

[Paul et al., 2015c] Paul, T., Puscher, D., Wilk, S., and Strufe, T. (2015c). Systematic, large-scale analysis on the feasibility of media prefetching in online social networks. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*.

[Paul et al., 2012c] Paul, T., Stopczynski, M., Puscher, D., Volkamer, M., and Strufe, T. (2012c). C4PS - helping Facebookers manage their privacy settings. In *SocInfo*.

[Perrin, 2003] Perrin, T. (2003). Public key distribution through "cryptoids". *Workshop on New Security Paradigms*.

[Puscher, 2014] Puscher, D. (2014). Sammlung und Analyse von Daten zum Nutzerverhalten in Online Social Networks. Master thesis.

[Raiciu et al., 2009] Raiciu, C. et al. (2009). Roar: Increasing the flexibility and performance of distributed search. In *SIGCOMM*.

[Raji et al., 2011] Raji, F., Miri, A., Jazi, M. D., and Malek, B. (2011). Online social network with flexible and dynamic privacy policies. In *CSSE*.

[Ratnasamy et al., 2001] Ratnasamy, S. et al. (2001). A scalable content-addressable network. In *ACM SIGCOMM*.

[Reynolds and Vahdat, 2003] Reynolds, P. and Vahdat, A. (2003). Efficient peer-to-peer keyword searching. In *Middleware*.

[Rosenblum, 2007] Rosenblum, D. (2007). What anyone can know: The privacy risks of social networking sites. *IEEE Security & Privacy*, 5(3).

[Rowstron and Druschel, 2001] Rowstron, A. and Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Middleware*.

[Rzadca et al., 2010] Rzadca, K. et al. (2010). Replica placement in p2p storage: Complexity and game theoretic analyses. In *IEEE ICDCS*.

[Sahai and Waters, 2005] Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. In *Advances in Cryptology – EUROCRYPT 2005*. Springer.

[Schneider et al., 2009] Schneider, F., Feldmann, A., Krishnmurthy, B., and Willinger, W. (2009). Understanding online social network usage from a network perspective. *IMC*.

[Schulz and Strufe, 2013] Schulz, S. and Strufe, T. (2013). d^2 deleting diaspora: Practical attacks for profile discovery and deletion. In *ICC*.

[Schwittmann et al., 2013] Schwittmann, L., Boelmann, C., Wander, M., and Weis, T. (2013). Sonet–privacy and replication in federated online social networks. In *ICDCSW*, pages 51–57. IEEE.

[Seong et al., 2010] Seong, S., Seo, J., Nasielsky, M., Sengupta, D., Hangal, S., Teh, S. K., Chu, R., Dodson, B., and Lam, M. S. (2010). Prpl: A decentralized social networking infrastructure. *Workshop of on Mobile Computing and Services: Social Networks and Beyond*.

[Shahriar et al., 2013] Shahriar, N., Chowdhury, S. R., Sharmin, M., Ahmed, R., Boutaba, R., and Mathieu, B. (2013). Ensuring beta-availability in p2p social networks. In *ICDCSW*.

[Shakimov et al., 2011] Shakimov, A. et al. (2011). Vis-a-vis: Privacy-preserving online social networking via virtual individual servers. In *COMSNETS*.

[Shakimov et al., 2009] Shakimov, A., Varshavsky, A., Landon, L. P., and Càceres, R. (2009). Privacy, cost, and availability tradeoffs in decentralized osns. *WOSN*.

[Sharma and Datta, 2012] Sharma, R. and Datta, A. (2012). Supernova: Super-peers based architecture for decentralized online social networks. In *COMSNETS*.

[Sharma et al., 2011] Sharma, R., Datta, A., DeH'Amico, M., and Michiardi, P. (2011). An empirical study of availability in friend-to-friend storage systems. In *IEEE P2P*.

[Simpson, 2008] Simpson, A. (2008). On the need for user-defined fine-grained access control policies for social networking applications. In *Workshop on Security in*

Opportunistic and SOCial networks. ACM.

[Steiner et al., 2007] Steiner, M., En-Najjary, T., and Biersack, E. W. (2007). A Global View of KAD. In *ACM SIGCOMM*.

[Steiner et al., 2009] Steiner, M., En-Najjary, T., and Biersack, E. W. (2009). Long term study of peer behavior in the kad dht. *IEEE/ACM TON*.

[Stoica et al., 2001] Stoica, I. et al. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM*.

[Strufe, 2010] Strufe, T. (2010). Profile popularity in a business-oriented online social network. In *Proceedings of the EuroSys Conference*.

[Stutzman et al., 2013] Stutzman, F., Gross, R., and Acquisti, A. (2013). Silent listeners: The evolution of privacy and disclosure on facebook. *Journal of Privacy and Confidentiality*.

[Sun et al., 2010] Sun, J., Zhu, X., and Fang, Y. (2010). A privacy-preserving scheme for online social networks with efficient revocation. In *INFOCOM*.

[Tegeler et al., 2011] Tegeler, F., Koll, D., and Fu, X. (2011). Gemstone: Empowering decentralized social networking with high data availability. In *GLOBECOM*.

[Tootoonchian et al., 2009] Tootoonchian, A., Saroiu, S., and A. Wolman, Y. G. (2009). Lockr: Better privacy for social network. *CoNEXT*.

[Ugander et al., 2011] Ugander, J., Karrer, B., Backstrom, L., and Marlow, C. (2011). The anatomy of the facebook social graph. *CoRR*.

[Varga and Hornig, 2008] Varga, A. and Hornig, R. (2008). An Overview of the OMNeT++ Simulation Environment. In *SIMUTools*.

[Velayathan and Yamada, 2007] Velayathan, G. and Yamada, S. (2007). Investigating user browsing behavior. In *WI-IAT*. IEEE.

[Wang et al., 2011] Wang, Z., Sun, L., and Yang, S. (2011). Prefetching strategy in peer-assisted social video streaming. In *ACM Conference on Multimedia*.

[Weinreich et al., 2006] Weinreich, H., Obendorf, H., Herder, E., and Mayer, M. (2006). Off the beaten tracks: exploring three aspects of web navigation. In *WWW*.

[Wilson et al., 2011] Wilson, C., Steinbauer, T., Wang, G., Sala, A., Zheng, H., and Zhao, B. Y. (2011). Privacy, availability and economics in the polaris mobile social network. In *HotMobile*.

[Wilson et al., 2012] Wilson, R. E., Gosling, S. D., and Graham, L. T. (2012). A review of facebook research in the social sciences. *Perspectives on psychological science*.

[Zhao et al., 2013] Zhao, Y., Do, N., Wang, S.-T., Hsu, C.-H., and Venkatasubramanian, N. (2013). O2sm: Enabling efficient offline access to online social media and social networks. In Eyers, D. and Schwan, K., editors, *Middleware 2013*, pages 445–465.

[Zych et al., 2015] Zych, I., Ortega-Ruiz, R., and Rey, R. D. (2015). Scientific research on bullying and cyberbullying: Where have we been and where are we going. *Aggression and Violent Behavior*.